

Globus Toolkit 4.2.1 Commandline Tools

Globus Toolkit 4.2.1 Commandline Tools

Abstract

You can also download a [PDF version here](#)¹.

¹ gtCommands.pdf

Table of Contents

| | |
|--|-----|
| I. Java WS Core Commands | ? |
| globus-start-container | 10 |
| globus-stop-container | 11 |
| globus-start-container-detached | 13 |
| globus-stop-container-detached | 14 |
| wsrf-destroy | 15 |
| wsrf-set-termination-time | 17 |
| wsrf-query | 19 |
| wsrf-get-property | 21 |
| wsrf-get-properties | 23 |
| wsrf-insert-property | 25 |
| wsrf-update-property | 27 |
| wsrf-delete-property | 29 |
| wsn-get-current-message | 31 |
| wsn-pause-subscription | 33 |
| wsn-resume-subscription | 35 |
| wsn-subscribe | 37 |
| globus-deploy-gar | 39 |
| globus-undeploy-gar | 41 |
| globus-check-environment | 43 |
| globus-check-remote-environment | 44 |
| globus-update-client-config | 45 |
| globus-validate-descriptors | 46 |
| globus-reload-container | 47 |
| globus-remote-undeploy-gar | 49 |
| globus-remote-deploy-gar | 51 |
| ws-enumerate-start | 53 |
| ws-enumerate | 55 |
| ws-enumerate-end | 57 |
| globus-xpath-query | 59 |
| Common Java Client Options | 62 |
| II. C WS Core Commands | 63 |
| globus-wsc-container | 64 |
| globus-wsrf-cgen | 66 |
| globus-wsrf-destroy | 70 |
| globus-wsrf-set-termination-time | 72 |
| globus-wsrf-query | 74 |
| globus-wsrf-get-property | 77 |
| globus-wsrf-get-properties | 79 |
| globus-wsrf-insert-property | 81 |
| globus-wsrf-update-property | 84 |
| globus-wsrf-delete-property | 87 |
| globus-wsn-get-current-message | 89 |
| globus-wsn-pause-subscription | 92 |
| globus-wsn-resume-subscription | 94 |
| globus-wsn-subscribe | 96 |
| III. GSI Commands | 99 |
| grid-cert-diagnostics | 100 |
| grid-cert-info | 102 |
| grid-cert-request | 104 |
| grid-default-ca | 107 |

| | |
|---|-----|
| grid-change-pass-phrase | 109 |
| grid-proxy-init | 110 |
| grid-proxy-destroy | 113 |
| grid-proxy-info | 114 |
| grid-mapfile-add-entry | 116 |
| grid-mapfile-check-consistency | 117 |
| grid-mapfile-delete-entry | 118 |
| IV. CAS Query Commands | ? |
| cas-whoami | 120 |
| cas-list-object | 122 |
| cas-get-object | 124 |
| cas-group-list-entries | 126 |
| cas-find-policies | 128 |
| query-cas-service | 130 |
| V. CAS Admin Commands | 132 |
| cas-proxy-init | 133 |
| cas-wrap | 136 |
| cas-enroll | 139 |
| cas-remove | 143 |
| cas-action | 146 |
| cas-group-admin | 148 |
| cas-group-add-entry | 152 |
| cas-group-remove-entry | 155 |
| cas-rights-admin | 158 |
| VI. Delegation Service Commands | 161 |
| globus-credential-delegate | 162 |
| globus-credential-refresh | 163 |
| globus-delegation-client | 165 |
| VII. GridFTP Commands | 168 |
| globus-url-copy | 169 |
| globus-gridftp-server | 179 |
| VIII. RFT Commands | 189 |
| rft | 190 |
| globus-crft | 192 |
| rft-delete | 194 |
| IX. Replica Location Service (RLS) Commands | 195 |
| globus-rls-admin | 196 |
| globus-rls-cli | 198 |
| globus-rls-server | 202 |
| X. WS RLS Commands | ? |
| globus-repicalocation-createmappings | 208 |
| globus-repicalocation-addmappings | 209 |
| globus-repicalocation-deletemappings | 210 |
| globus-repicalocation-defineattributes | 211 |
| globus-repicalocation-undefineattributes | 212 |
| globus-repicalocation-addattributes | 213 |
| globus-repicalocation-modifyattributes | 214 |
| globus-repicalocation-removeattributes | 215 |
| XI. DataRep Commands | ? |
| globus-replication-create | 217 |
| globus-replication-start | 219 |
| globus-replication-stop | 220 |
| globus-replication-suspend | 221 |
| globus-replication-resume | 222 |

Globus Toolkit 4.2.1 Commandline
Tools

| | |
|---|-----|
| globus-replication-finditems | 223 |
| XII. Replication Client Commands | ? |
| globus-replication-client | 225 |
| XIII. WS MDS Commands | 226 |
| mds-servicegroup-add | 227 |
| mds-set-multiple-termination-time | 230 |
| XIV. GRAM4 Commands | 231 |
| globusrun-ws | 232 |
| XV. GridWay Commands | 239 |
| Job and Array Job submission Command | 240 |
| DAG Job submission Command | 241 |
| Job Monitoring Command | 242 |
| Job History Command | 244 |
| Host Monitoring Command | 245 |
| Job Control Command | 247 |
| Job Synchronization Command | 248 |
| User Monitoring Command | 249 |
| Accounting Command | 250 |
| JSDL To GridWay Job Template Parser Command | 251 |
| Glossary | 252 |

List of Figures

| | |
|--|-----|
| 1. Effect of Parallel Streams in GridFTP | 177 |
|--|-----|

List of Tables

| | |
|---|----|
| 1. Options | 10 |
| 2. Common options | 12 |
| 3. Shutdown options | 12 |
| 4. Options | 13 |
| 5. Options | 14 |
| 6. Common options | 16 |
| 7. Command-specific options | 17 |
| 8. Common options | 18 |
| 9. Common options | 20 |
| 10. Common options | 22 |
| 11. Common options | 24 |
| 12. Common options | 26 |
| 13. Common options | 28 |
| 14. Common options | 30 |
| 15. Common options | 32 |
| 16. Common options | 34 |
| 17. Common options | 36 |
| 18. Command-specific options | 37 |
| 19. Common options | 38 |
| 20. Options | 39 |
| 21. Supported property-value pairs | 39 |
| 22. Options | 41 |
| 23. Options | 44 |
| 24. Options | 45 |
| 25. Options | 46 |
| 26. Common options | 48 |
| 27. Common options | 50 |
| 28. Command-specific options | 51 |
| 29. Common options | 52 |
| 30. Common options | 54 |
| 31. Command-specific options | 55 |
| 32. Common options | 56 |
| 33. Common options | 58 |
| 34. Command-specific options | 59 |
| 35. Common options | 60 |
| 36. Common options | 62 |
| 37. WSRF Core Namespaces and C Prefixes | 68 |
| 38. Common options | 70 |
| 39. Common options | 72 |
| 40. Application-specific options | 74 |
| 41. Common options | 75 |
| 42. Common options | 77 |
| 43. Common options | 79 |
| 44. Common options | 82 |
| 45. Common options | 85 |
| 46. Common options | 87 |
| 47. Application-specific options | 89 |
| 48. Common options | 90 |
| 49. Common options | 92 |
| 50. Common options | 94 |
| 51. Application-specific options | 96 |

| | |
|--|-----|
| 52. Common options | 97 |
| 53. Command line options | 109 |
| 54. Command line options | 111 |
| 55. Command line options | 113 |
| 56. Command line options | 114 |
| 57. Print options | 114 |
| 58. Validity options | 115 |
| 59. Command line options | 116 |
| 60. Command line options | 117 |
| 61. Command line options | 118 |
| 62. globus-credential-delegate options | 162 |
| 63. globus-credential-refresh options | 164 |
| 64. Common options | 166 |
| 65. Application-specific options | 166 |
| 66. URL formats | 171 |
| 67. Options for globus-rls-admin | 197 |
| 68. Options for globus-rls-cli | 198 |
| 69. Commands for globus-rls-cli | 200 |
| 70. Options for globus-rls-server | 205 |
| 71. globus-repicalocation-createmappings Options | 208 |
| 72. globus-repicalocation-addmappings Options | 209 |
| 73. globus-repicalocation-deletemappings Options | 210 |
| 74. globus-repicalocation-defineattributes Options | 211 |
| 75. globus-repicalocation-undefineattributes Options | 212 |
| 76. globus-repicalocation-addattributes Options | 213 |
| 77. globus-repicalocation-modifyattributes Options | 214 |
| 78. globus-repicalocation-removeattributes Options | 215 |
| 79. Options | 218 |
| 80. Options | 219 |
| 81. Options | 220 |
| 82. Options | 221 |
| 83. Options | 222 |
| 84. Options | 223 |
| 85. Options | 225 |
| 86. Commands And Arguments | 225 |
| 87. Field options | 243 |
| 88. Field information | 244 |
| 89. Field information | 245 |
| 90. Queue field information | 246 |
| 91. Field information | 249 |
| 92. Field information | 250 |

Java WS Core Commands

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 `wsrf-*` and `wsn-*` clients should work against any service that supports the given WSRF or WSN operations.

Name

`globus-start-container --` Starts standalone container

`globus-start-container`

Tool description

Starts a standalone container. By default a secure container is started on port 8443 and is accessible via HTTPS. On successful startup a list of services will be displayed on the console. By default the non secure (HTTP) container is started on port 8080.

Command syntax

```
globus-start-container [options]
```

Table 1. Options

| | |
|------------------------------------|--|
| -help | Displays help information about the command. |
| -p <port> | Sets the port number for the container. |
| -i <address> | Binds container to the specified network address. |
| -quiet | Does not show a list of services at startup. |
| -debug | Enables debug mode. |
| -nosec | Starts a non secure (HTTP) container. Please note that this option only disables transport security. Message security can still be used. |
| -containerDesc <file> | Specifies a container security descriptor file. |
| -profile <name> | Specifies a configuration profile name for the container. |

Name

`globus-stop-container --` Stops standalone container

`globus-stop-container`

Tool description

Stops a standalone container. By default this command will attempt to stop a container running on **localhost:8443** and perform a **soft** shutdown.

The **globus-stop-container** command invokes a **ShutdownService** running in the container. By default that service is configured to perform **self** authorization and therefore the **globus-stop-container** must be executed with the same credentials as the container it is running with. Alternatively, the service can be configured with a gridmap file to allow a subset of users (with their own credentials) to invoke the service (please see the service security deployment descriptor for details).

Command syntax

```
globus-stop-container [options] ['soft' | 'hard']
```

Table 2. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Table 3. Shutdown options

| | |
|---------------|---|
| 'soft' | This option lets the threads die naturally. |
| 'hard' | This option forces an immediate JVM shutdown. |

Example:

```
$ globus-stop-container soft
```

Please see the [troubleshooting section](#) if you are having problems with `globus-stop-container`.

Name

`globus-start-container-detached --` Starts standalone container detached from controlling TTY

`globus-start-container-detached`

Tool description

Starts a standalone container detached from the controlling TTY. This can be useful for long running containers or when started from `init.d` scripts. Container log goes to `$GLOBUS_LOCATION/var/container.log` and a PID file is written to `$GLOBUS_LOCATION/var/container.pid`. `globus-start-container-detached` is just a wrapper around `globus-start-container` so see [globus-start-container](#) for more information and options.



Note

Note that this tool is only available after doing a full Globus install. It is not available in Java WS Core only installs.

Command syntax

```
globus-start-container-detached [options] | [arguments to container]
```

Table 4. Options

| | |
|------------------------------|--|
| -logfile <file> | Specifies an alternate container log file. |
| -append | Do not overwrite the existing log file. |
| -pidfile <file> | Specifies an alternate PID file location. |

Name

`globus-stop-container-detached --` Stops standalone container detached from controlling TTY

`globus-stop-container-detached`

Tool description

Stops a standalone container detached from the controlling TTY. `$GLOBUS_LOCATION/var/container.pid` is used to find the PID of the running container and signals are sent to stop the container.



Note

Note that this tool is only available after doing a full Globus install. It is not available in Java WS Core only installs.

Command syntax

```
globus-stop-container-detached [options]
```

Table 5. Options

| | |
|------------------------------------|---|
| <code>-pidfile <file></code> | Specifies an alternate PID file location. |
|------------------------------------|---|

Name

`wsrf-destroy --` Destroys a resource

`wsrf-destroy`

Tool description

Destroys a resource.

Command syntax

`wsrf-destroy [options]`

Table 6. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either ' limited ' or ' full '. Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be ' msg ' for GSI Secure Message, or ' conv ' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be ' sig ' for signature or ' enc ' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be ' self ', ' host ', ' none ', or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ wsrfl-destroy -s http://localhost:8080/wsrfl/services/CounterService \ -k
  "{http://counter.com}CounterKey" 123
```

Name

`wsrf-set-termination-time` -- Sets termination time of a resource

`wsrf-set-termination-time`

Tool description

Sets a termination time of a resource.

Command syntax

`wsrf-set-termination-time` [options] <seconds> | 'infinity'

The following are command-specific options in addition to the common options:

Table 7. Command-specific options

| | |
|------------------|----------------------|
| -u, --utc | Display time in UTC. |
|------------------|----------------------|

Table 8. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either ' limited ' or ' full '. Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be ' msg ' for GSI Secure Message, or ' conv ' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be ' sig ' for signature or ' enc ' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be ' self ', ' host ', ' none ', or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ wsrif-set-termination-time -s http://localhost:8080/wsrif/services/CounterService \ -k
  "{http://counter.com}CounterKey" 123 30
```

Name

wsrf-query -- Performs query on a resource property document

wsrf-query

Tool description

Queries the resource property document of a resource. By default, a simple XPath query is assumed that returns the entire resource property document.

Command syntax

```
wsrf-query [options] [query expression] [dialect]
```

Table 9. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Examples:

```
$ wsrif-query -s https://127.0.0.1:8443/wsrif/services/DefaultIndexService \
  "count(//*[local-name()='Entry'])"
```

```
$ wsrif-query -s https://127.0.0.1:8443/wsrif/services/DefaultIndexService \
  "number(//*[local-name()='GLUECE']/glue:ComputingElement/glue:State/@glue:FreeCPUs)=0"
```

```
$ wsrif-query -s http://localhost:8080/wsrif/services/ContainerRegistryService \
  "/*/*/*/*[local-name()='Address']"
```

Name

`wsrp-get-property --` Gets values of a single resource property

`wsrp-get-property`

Tool description

Gets a single resource property from a resource.

Command syntax

`wsrp-get-property [options] <property>`

The `<property>` is a QName of the resource property in the string form: `{namespaceURI}localPart`.

Table 10. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ wsrif-get-property -s http://localhost:8080/wsrif/services/CounterService \ -k
  "{http://counter.com}CounterKey" 123 \
  "{http://docs.oasis-open.org/wsrif/2004/06/wsrif-WS-ResourceLifetime-1.2-draft-01.xsd}Cu
```

Name

`wsrf-get-properties --` Gets values of multiple resource properties

`wsrf-get-properties`

Tool description

Gets multiple resource properties from a resource.

Command syntax

```
wsrf-get-properties [options] <property1> [<property2>...  
  <propertyN>]
```

Each **<propertyN>** is a QName of the resource property in the string form: **{namespaceURI}localPart**.

Table 11. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ wsrfg-get-properties -s http://localhost:8080/wsrfg/services/CounterService \ -k
  "{http://counter.com}CounterKey" 123 \
  "{http://docs.oasis-open.org/wsrfg/2004/06/wsrfg-WS-ResourceLifetime-1.2-draft-01.xsd}Cu
  \
  "{http://docs.oasis-open.org/wsrfg/2004/06/wsrfg-WS-ResourceLifetime-1.2-draft-01.xsd}Te
```

Name

wsrc-insert-property -- Inserts values into a resource property

wsrc-insert-property

Tool description

Inserts a property into the resource property document of a resource.

Command syntax

```
wsrc-insert-property [options] <propertyValueFile>
```

The **<propertyValueFile>** is an XML file that contains the value of the resource property. The QName of the property is the outer most element.

Table 12. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: <pre>-k "{http://www.globus.org}MyKey" 123</pre> |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example: Contents of **in.xml**:

```
<doc> <ns1:foo xmlns:ns1="http://widgets.com"> Value1
  </ns1:foo> <ns1:foo xmlns:ns1="http://widgets.com"> Value2
</ns1:foo> </doc>
```

```
$ wsrfl-insert-property -s http://localhost:8080/wsrfl/services/WidgetService \ -k
  "{http://www.globus.org/namespaces/2004/06/core}WidgetKey" 123 \ in.xml
```

Name

`wsrfr-update-property` -- Updates value of a resource property

`wsrfr-update-property`

Tool description

Updates the property value in the resource property document of a resource.

Command syntax

```
wsrfr-update-property [options] <propertyValueFile>
```

The **<propertyValueFile>** is an XML file that contains the value of the resource property. The QName of the property is the outermost element.

Table 13. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example: Contents of **in.xml**:

```
<doc> <ns1:foo xmlns:ns1="http://widgets.com"> Value
  </ns1:foo> </doc>
```

```
$ wsrfe-update-property -s http://localhost:8080/wsrfe/services/WidgetService \ -k
  "{http://www.globus.org/namespaces/2004/06/core}WidgetKey" 123 \ in.xml
```

Name

wsrf-delete-property -- Deletes a resource property

wsrf-delete-property

Tool description

Deletes a resource property from the resource property document of a resource.

Command syntax

```
wsrf-delete-property [options] <property>
```

The <property> is a QName of the resource property in the string form: **{namespaceURI}localPart**.

Table 14. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ wsrfe-delete-property -s http://localhost:8080/wsrfe/services/WidgetService \ -k
  "{http://www.globus.org/namespaces/2004/06/core}WidgetKey" 123 \
  "{http://widgets.com}foo"
```

Name

`wsn-get-current-message --` Gets a current message associated with a topic

`wsn-get-current-message`

Tool description

Gets the current message associated with the specified topic.

Command syntax

`wsn-get-current-message [options] <topic>`

The `<topic>` is a QName of the resource property in the string form: `{namespaceURI}localPart`.

Table 15. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: <pre>-k "{http://www.globus.org}MyKey" 123</pre> |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ wsn-get-current-message -s
  http://localhost:8080/wsrf/services/CounterService \ -k "{http://counter.com}CounterK
  "{http://counter.com}Value"
```

Name

`wsn-pause-subscription --` Pauses a subscription

`wsn-pause-subscription`

Tool description

Pauses a subscription (notifications on that subscription will not be sent out until it is resumed).

Command syntax

`wsn-pause-subscription [options]`

Table 16. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ wsn-pause-subscription -s
  http://localhost:8080/wsrf/services/SubscriptionManagerService \ -k
  "{http://www.globus.org/namespaces/2004/06/core}acc271c0-4df9-11d9-ab19-87da3bc7cf28"
```

Name

`wsn-resume-subscription --` Resumes a subscription

`wsn-resume-subscription`

Tool description

Resumes a subscription (notifications on that subscription will be sent out again).

Command syntax

`wsn-resume-subscription [options]`

Table 17. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either ' limited ' or ' full '. Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be ' msg ' for GSI Secure Message, or ' conv ' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be ' sig ' for signature or ' enc ' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be ' self ', ' host ', ' none ', or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ wsn-resume-subscription -s
  http://localhost:8080/wsrf/services/SubscriptionManagerService \ -k
  "{http://www.globus.org/namespaces/2004/06/core}acc271c0-4df9-11d9-ab19-87da3bc7cf28"
```

Name

wsn-subscribe -- Subscribes to a topic

wsn-subscribe

Tool description

Subscribes to a topic.

Command syntax

```
wsn-subscribe [options] <topic>
```

The <topic> is a QName of the resource property in the string form: **{namespaceURI}localPart**.

The following are subscribe-specific options in addition to the common options:

Table 18. Command-specific options

| | |
|---------------------------------------|--|
| -r, --resDescFile <file> | Specifies a file containing a resource security descriptor for the notification consumer resource. |
| -b, --subEpr <file> | Specifies a file to which the subscription resource EPR will be saved. |

Table 19. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ wsn-subscribe -s http://localhost:8080/wsrf/services/CounterService \ -k
  "{http://counter.com}CounterKey" 123 \ "{http://counter.com}Value"
```

Name

globus-deploy-gar -- Deploys a GAR file (locally)

globus-deploy-gar

Tool description

Deploys a GAR file (locally) into Java WS Core or Apache Tomcat container.

Command syntax

```
globus-deploy-gar [options] <gar.file>
```

The <gar.file> is the path to the GAR file to be deployed.

Table 20. Options

| | |
|---|--|
| -help | Displays help information about the command. |
| -debug | Enables debug mode. |
| -verbose | Enables verbose mode. |
| -backup | Creates backup of existing configuration files. |
| -overwrite | Overwrite existing deployment. |
| -profile <name> | Specifies the profile name under which the configuration files in the GAR will be deployed. Please see "Configuration Profiles" under Configuring Java WS Core for details. |
| -tomcat <dir> | Deploys a GAR file to Apache Tomcat. The <dir> argument must point to the Tomcat installation directory. <i>Note:</i> Java WS Core must be already deployed in Tomcat. Please see Deploying into Tomcat section for details. |
| -D<property>=<value> | Passes arbitrary property-value pairs. See below for the list of currently supported properties . |

Table 21. Supported property-value pairs

| | |
|-----------------------------|--|
| -Dall.scripts=true | Causes Windows and Unix launcher scripts to be generated. |
| -DdoValidation=false | Turns off automatic validation of service configuration files. |



Note

Since GT 4.2, **globus-deploy-gar** command will NOT overwrite the existing deployment unless `-overwrite` option is specified. It is recommended to undeploy the existing deployment first. The container must be off to deploy a GAR file.

Example I:

```
$ globus-deploy-gar /tmp/gars/globus_wsrf_core_samples_counter.gar
```

The above command will deploy the `globus_wsrf_core_samples_counter.gar` into Java WS Core installation directory. The above command invokes the `deployGar` task in the `build-packages.xml` Ant build file. The above example is equivalent to running:

```
$ ant -f $GLOBUS_LOCATION/share/globus_wsrf_common/build-packages.xml deployGar \  
-Dgar.name=/tmp/gars/globus_wsrf_core_samples_counter.gar
```

The profile name can be passed using the **-Dprofile** Ant option. To enable back up of the existing configuration files add the **-DcreateBackup=true** Ant option. Make sure to use the *absolute* path name for the `gar` file when using Ant directly.

Example II:

```
$ globus-deploy-gar -tomcat /soft/tomcat-5.5.20 \  
/tmp/gars/globus_wsrf_core_samples_counter.gar
```

The above command will deploy the `globus_wsrf_core_samples_counter.gar` into Apache Tomcat. The above command invokes the `deployGar` task in the `tomcat-service.xml` Ant build file. The above example is equivalent to running:

```
$ ant -f $GLOBUS_LOCATION/share/globus_wsrf_common/tomcat/tomcat-service.xml deployGar \  
-Dgar.name=/tmp/gars/globus_wsrf_core_samples_counter.gar \ -Dtomcat.dir=/soft/tomcat-
```

By default the GAR file will be deployed under the "wsrf" web application. To specify a different web application name use **-Dwebapp.name=<name>** option.

Name

globus-undeploy-gar -- Undeploys a GAR file (locally)

globus-undeploy-gar

Tool description

Undeploys a GAR file (locally) from Java WS Core or Apache Tomcat container.

Command syntax

```
globus-undeploy-gar [options] <gar.id>
```

The <gar.id> is the base name of the GAR file without the **.gar** extension to undeploy. For example if the GAR file is "foo.gar", then the GAR id is "foo". The directory names under **\$GLOBUS_LOCATION/etc/globus_packages/** are the GAR ids of the undeployable items.

Table 22. Options

| | |
|---|---|
| -help | Displays help information about the command. |
| -debug | Enables debug mode. |
| -verbose | Enables verbose mode. |
| -tomcat <dir> | Undeploy a GAR file from Apache Tomcat. The <dir> argument must point to the Tomcat installation directory. |
| -D<property>=<value> | Passes arbitrary property-value pairs. |



Note

The container must be off to undeploy a GAR file.

Example I:

```
$ globus-undeploy-gar globus_wsrf_core_samples_counter
```

The above command will undeploy `globus_wsrf_core_samples_counter` GAR from Java WS Core installation directory. The above command invokes the **undeployGar** task in the `build-packages.xml` Ant build file. The above example is equivalent to running:

```
$ ant -f $GLOBUS_LOCATION/share/globus_wsrf_common/build-packages.xml undeployGar \
-Dgar.id=globus_wsrf_core_samples_counter
```

Example II:

```
$ globus-undeploy-gar -tomcat /soft/tomcat-5.5.20 \ globus_wsrf_core_samples_counter
```

The above command will undeploy `globus_wsrf_core_samples_counter` GAR from Apache Tomcat. The above command invokes the **undeployGar** task in the `tomcat-service.xml` Ant build file. The above example is equivalent to running:

```
$ ant -f $GLOBUS_LOCATION/share/globus_wsrf_common/tomcat/tomcat-service.xml undeployGar \
-Dgar.id=globus_wsrf_core_samples_counter \ -Dtomcat.dir=/soft/tomcat-5.5.20
```

By default the GAR file will be undeployed under the "wsrf" web application. To specify a different web application name use **-Dwebapp.name=<name>** option.

Name

globus-check-environment -- Displays component version information and validates JVM version.

globus-check-environment

Tool description

Displays component version information and validates the JVM version. This tool is primarily used for debugging purposes.

Name

globus-check-remote-environment -- Displays remote component version information.

globus-check-remote-environment

Tool description

Displays remote component version information.

Command syntax

```
globus-check-environment [-help] -s endpoint -z authz
```

Table 23. Options

| | |
|--------------------|---|
| -help | Displays help information about the command. |
| -s endpoint | Remote endpoint to print version information about. It should be of the format protocol://host:port, example https://localhost:8443. |
| -z authz | Sets authorization, can be 'self', 'host', 'hostOrSelf' or 'none' or a string specifying the expected identity of the remote party. Defaults to no authorization. |

Name

`globus-update-client-config --` Merges `client-config.wsdd` files from deployed modules into the global `client-config.wsdd` configuration file

`globus-update-client-config`

Tool description

Merges multiple `client-config.wsdd` files from deployed modules into the global configuration file. Scans each `$GLOBUS_LOCATION/etc/<modulename>/client-config.wsdd` and merges the contents into `$GLOBUS_LOCATION/client-config.wsdd`. This tool is primarily intended for use by administrators and automation tools to facilitate the adding and removing of module specific type-mapping and/or other client-side configuration from the global `client-config.wsdd` file used by the Globus installation.

Command syntax

```
globus-update-client-config [<filename>]
```

Table 24. Options

| | |
|-------------------------|---|
| <filename> | Optional argument that specifies an alternate path to write the result <code>client-config.wsdd</code> file. By default, running the program with no arguments will write the file to <code>\$GLOBUS_LOCATION/client-config.wsdd</code> |
|-------------------------|---|

Name

globus-validate-descriptors -- Validate configuration files of all services

globus-validate-descriptors

Tool description

Validates the Web Services Deployment Descriptor (.wsdd) files, JNDI configuration files (jndi-config.xml), and security descriptors for all services.

Command syntax

globus-validate-descriptors [options]

Table 25. Options

| | |
|---|--|
| -help | Displays help information about the command. |
| -debug | Enables debug mode. |
| -verbose | Enables verbose mode. |
| -D<property>=<value> | Passes arbitrary property-value pairs. |

Name

`globus-reload-container --` Reinitializes standalone container

`globus-reload-container`

Tool description

Invokes the `reload()` operation on the **DeployService** running in the remote container. It tells the container to reinitialize all of its services, re-read its and service configuration files, etc. For example, the administrator can change the security descriptor of a service and then use the **globus-reload-container** command to force the container to load the updated configuration without restarting the container.

By default the **DeployService** is configured to perform **self** authorization and therefore the **globus-reload-container** must be executed with the same credentials as the container it is running with. Alternatively, the service can be configured with a gridmap file to allow a subset of users (with their own credentials) to invoke the service (please see the service security deployment descriptor for details).



Note

This command only works with the standalone container. Please see the [Java WS Core Dynamic Deploy Design Document](#) for more information.

Command syntax

`globus-reload-container [options]`

Table 26. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: <pre>-k "{http://www.globus.org}MyKey" 123</pre> |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ globus-reload-container
```

Name

globus-remote-undeploy-gar -- Undeploys a GAR file (remotely)

globus-remote-undeploy-gar

Tool description

The **globus-remote-undeploy-gar** command undeploys a GAR file remotely. It invokes the **undeploy()** operation on the **DeployService** running in the remote container. It works just like the [globus-undeploy-gar](#) command but the GAR file is undeployed remotely.

By default the **DeployService** is configured to perform **self** authorization and therefore the **globus-remote-undeploy-gar** must be executed with the same credentials as the container it is running with. Alternatively, the service can be configured with a gridmap file to allow a subset of users (with their own credentials) to invoke the service (please see the service security deployment descriptor for details).



Note

This command only works with the standalone container. Please see the [Java WS Core Dynamic Deploy Design Document](#) for more information.

Command syntax

```
globus-remote-undeploy-gar [options] <gar.id>
```

The **<gar.id>** is the base name of the GAR file without the **.gar** extension to undeploy. For example if the GAR file is **"foo.gar"**, then the GAR id is **"foo"**.

Table 27. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ globus-remote-undeploy-gar globus_wsrf_core_samples_counter
```

To see what GAR files can be undeployed on the remote container run the following query on the **DeployService**, for example:

```
$ wsrf-query -z hostSelf -s https://127.0.0.1:8443/wsrf/services/DeployService
```

Name

globus-remote-deploy-gar -- Deploys a GAR file (remotely)

globus-remote-deploy-gar

Tool description

The **globus-remote-deploy-gar** command deploys a GAR file remotely. It first transfers the GAR file to the **DeployService** running in the remote container and then it deploys it using the **deploy()** operation of the service (the tool can also perform these two operations separately).

By default the **DeployService** is configured to perform **self** authorization and therefore the **globus-remote-deploy-gar** must be executed with the same credentials as the container it is running with. Alternatively, the service can be configured with a gridmap file to allow a subset of users (with their own credentials) to invoke the service (please see the service security deployment descriptor for details).



Note

This command only works with the standalone container. Please see the [Java WS Core Dynamic Deploy Design Document](#) for more information.

Command syntax

```
globus-remote-deploy-gar [options] <gar>
```

The **<gar>** can be either an URL or a file location. If a file location is passed to the tool, it will transfer the file to the service via SOAP with Attachments (the **upload()** function) using the **MTOM** format. If an URL is passed, the tool will call the **download()** function of the service, and let the service download the GAR file.

The following are command-specific options in addition to the common options:

Table 28. Command-specific options

| | |
|------------------------|---|
| -n, --transfer | Transfer GAR file only. |
| -y, --deploy | Deploy GAR file only (assumes the GAR is already transferred to the DeployService). |
| -o, --overwrite | Overwrite existing deployment. |
| -b, --backup | Creates backup of existing configuration files |

Table 29. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Examples:

```
$ globus-remote-deploy-gar /tmp/myService.gar
```

```
$ globus-remote-deploy-gar gsiftp://localhost/tmp/myService.gar
```

To see what GAR files have been transferred but not yet deployed on the remote container run the following query on the **DeployService**, for example:

```
$ wsrfe-query -z hostSelf -s https://127.0.0.1:8443/wsrfe/services/DeployService
```

Name

`ws-enumerate-start --` Starts an enumeration

`ws-enumerate-start`

Tool description

Creates a new enumeration context and prints it out to the console.



Note

The remote service must support the **enumerate** operation of the WS-Enumeration specification.

Command syntax

`ws-enumerate-start [options]`

Table 30. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either ' limited ' or ' full '. Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be ' msg ' for GSI Secure Message, or ' conv ' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be ' sig ' for signature or ' enc ' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be ' self ', ' host ', ' none ', or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ ws-enumerate-start -s http://localhost:8080/wsrfl/services/ContainerRegistryService \
> enum.context
```

The created enumeration context will be stored in the `enum.context` file which then can be passed to `ws-enumerate` and `ws-enumerate-end` command line clients.

Name

ws-enumerate -- Retrieves enumeration data

ws-enumerate

Tool description

Retrieves the next set of enumeration data and prints it out to the console.



Note

The remote service must implement the WS-Enumeration specification.

Command syntax

```
ws-enumerate [options] <enumContextFile>
```

The **<enumContextFile>** is a file that contains the enumeration context.

The following are command-specific options in addition to the common options:

Table 31. Command-specific options

| | |
|--|---|
| -i, --items <int> | Specifies the total number of enumeration items to retrieve. The parameter value can be 'all' to retrieve the all the enumeration data. By default, only one element is retrieved. |
| -r, --maxCharacters <int> | Specifies the maximum number of characters (in Unicode) of the enumeration data that the client can accept at a time. By default, there is no limit on the size of the elements. |
| -n, --maxElements <int> | Specifies the maximum number of enumeration items to fetch at a time. By default, one element is retrieved at a time. |
| -o, --maxTime <int> | Specifies the maximum amount of time (in milliseconds) in which the enumeration data must be assembled. By default, there is no time limit. |

Table 32. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ ws-enumerate -s http://localhost:8080/wsrf/services/ContainerRegistryService \ -i 10
-n 5 enum.context
```

This command will display 10 elements of the enumeration data obtaining 5 elements at a time from the service.

Name

`ws-enumerate-end --` Stops an enumeration

`ws-enumerate-end`

Tool description

Releases an enumeration context.



Note

The remote service must implement the WS-Enumeration specification.

Command syntax

`ws-enumerate-end [options] <enumContextFile>`

The `<enumContextFile>` is a file that contains the enumeration context.

Table 33. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either ' limited ' or ' full '. Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be ' msg ' for GSI Secure Message, or ' conv ' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be ' sig ' for signature or ' enc ' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be ' self ', ' host ', ' none ', or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Example:

```
$ ws-enumerate-end -s http://localhost:8080/wsrf/services/ContainerRegistryService \
enum.context
```

Name

globus-xpath-query -- Performs XPath query on a resource property document

globus-xpath-query

Tool description

The **globus-xpath-query** uses a custom query dialect implementation called TargetedXPath to query the resource property document of a resource. Please see the [querying resource properties using XPath](#) section for more details.

Command syntax

```
globus-xpath-query [options] [query expression] [rpQName]
```

The **query expression** is an XPath expression. The **rpQName** is a resource property QName. If a resource property is specified only that resource property within the resource property document will be queried. Otherwise, the entire resource property document will be queried. By default, a simple XPath query is assumed that returns the entire resource property document.

Table 34. Command-specific options

| | |
|-------------------------------------|--|
| -n, --nsMapFile <file> | Specifies a file that contains namespace mappings. By default, the <code>etc/globus_wsrf_core/namespace-mappings.xml</code> file is used. |
| -u, --enumerate | Enumerate the query results. The query response will contain an enumeration context through which the actual query results can be obtained. The returned enumeration context can be used with the ws-enumerate command line tool. Also, please note that by default the enumeration context will expire in 30 minutes. |

Table 35. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: -k "{http://www.globus.org}MyKey" 123 |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

Examples:

```
$ globus-xpath-query -s http://localhost:8080/wsrfl/services/ContainerRegistryService \
  "/wssg:MemberServiceEPR/wsa:Address"
```

The above command will query the entire resource property document of the service.

```
$ globus-xpath-query -s http://localhost:8080/wsrfl/services/ContainerRegistryService \
  "/wssg:MemberServiceEPR/wsa:Address" wssg:Entry
```

The above command will query only the `wssg:Entry` resource property of the resource property document of the service.

```
$ globus-xpath-query -s http://localhost:8080/wsrf/services/ContainerRegistryService \  
-u "//wssg:MemberServiceEPR/wsa:Address" > enum.context $ ws-enumerate \  
-s http://localhost:8080/wsrf/services/ContainerRegistryService \ -i all enum.context
```

The first command will create an enumeration for the query results and store the returned enumeration context in a file. The second command will use the enumeration context stored in that file to retrieve the actual query results.

Name

Common Java Client Options -- list of common options across commands

Common Java Client Options

Table 36. Common options

| | |
|---|--|
| -h, --help | Displays help information about the command. |
| -d, --debug | Enables debug mode. For example, full stack traces of errors will be displayed. |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. |
| -s, --service <url> | Specifies the service URL. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , while the value is the simple value of the key. For complex keys, use the --eprFile option. Example: <pre>-k "{http://www.globus.org}MyKey" 123</pre> |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -g, --delegation <mode> | Enables delegation. mode can be either 'limited' or 'full' . Only supported with the GSI Secure Conversation authentication mechanism. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be 'msg' for GSI Secure Message, or 'conv' for GSI Secure Conversation. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -p, --protection <type> | Specifies the protection level. type can be 'sig' for signature or 'enc' for encryption. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z, --authorization <type> | Specifies authorization type. type can be 'self' , 'host' , 'none' , or a string specifying the expected identity of the remote party. |
| -t, --timeout <timeout> | Specifies client timeout (in seconds). The client will wait maximum of the timeout value for a response from the server before returning an error. By default the timeout value is 10 minutes. |

C WS Core Commands

Name

globus-wsc-container -- Host C web services

```
globus-wsc-container [-help] [-usage] [-version]
[-bg] [-pidfile PID]
[-max MAX-SESSIONS]
[-port PORT]
[-log LOGPATH]
[-nosec]
```

Description

The **globus-wsc-container** is a stand-alone SOAP service hosting container. It listens for SOAP / HTTP operation requests on a network port and dispatches those to dynamically loaded service modules. By default, **globus-wsc-container** will process SOAP messages until it receives a SIGINT or SIGTERM signal. In interactive usage, it typically runs until the user enters **Ctrl+C** on the keyboard.

The full set of command-line options to **globus-wsc-container** are:

| | |
|--------------------------|--|
| -help | Display a help message and exit |
| -usage | Display a short usage message and exit |
| -version | Display the program version and exit |
| -bg | Run the program as a daemon |
| -pid <i>PIDFILE</i> | Write the process ID of the program to <i>PIDFILE</i> |
| -max <i>MAX-SESSIONS</i> | Allow at most <i>MAX-SESSIONS</i> concurrent sessions to be processed by the program |
| -port <i>PORT</i> | Listen for SOAP/HTTP(s) connections on TCP port <i>PORT</i> |
| -log <i>LOGPATH</i> | Log container information to <i>LOGPATH</i> |
| -nosec | Disable TLS |

By default, the **globus-wsc-container** program picks an anonymous TCP port within values specified by the `GLOBUS_TCP_PORT_RANGE` environment variable, if present. To choose a specific port to listen on, pass the option `-port PORT` on the command-line of the process.

The **globus-wsc-container** program can also be run in the background as a daemon. This is done by passing the `-bg` command-line option. This can be combined with the `-pidfile PID` option to run in the background and record the PID of the process in a file, so that the daemon can be easily terminated.

By default, the container uses TLS for SOAP requests over https. This can be disabled to use unprotected http by passing the `-nosec` command-line option to this program. Message-level security may be enabled on a per-service basis if this is used.

To enable CEDPs "best practices" logging, pass the `-log LOGPATH` option to the container. The log file will contain name=value pairs for all events that the container processes.

By default the container will accept as many SOAP connections as the operating system will allow. To throttle the number of outstanding connections that can be processed in parallel, use the `-max MAX-SESSIONS` command-line option.

Services

The container looks for services in dynamic modules located in the `$GLOBUS_LOCATION/lib/globus_service_modules` directory. The Globus Toolkit ships with a number of sample services, test services, and implementations of the core *WSRF* services for implementing Resource Properties, Resource Lifetime, Service Groups, and Notifications. The **globus-wsrf-cgen** command parses *WSDL* schemas and generates service skeletons which can be used to implement additional web services.

Examples

Start a container in the foreground on port 8443:

```
% globus-wsc-container -port 8443
```

Contact: <https://grid.example.org:8443/>

Star a container as a daemon on an anonymous port, with a maximum of 64 parallel sessions, recording the port number to a file and logging to another file.

```
% globus-wsc-container \
  -bg \
  -pidfile $GLOBUS_LOCATION/var/globus-wsc-container.pid \
  -log $GLOBUS_LOCATION/var/globus-wsc-container.log \
  -max 64
  > $GLOBUS_LOCATION/var/globus-wsc-container.contact
```

```
% cat $GLOBUS_LOCATION/var/globus-wsc-container.contact
```

Contact: <https://grid.example.org:18332/>

```
% cat $GLOBUS_LOCATION/var/globus-wsc-container.log
```

```
ts=2008-06-19T22:43:21.645807Z id=21475 event=globus_service_engine.start engine_id=40235
```

Name

globus-wsrf-cgen -- Generate Stubs/Skeletons in C

```
globus-wsrf-cgen [-help] [-dr]
[-s PACKAGE-NAME] [-sn SERVICE-NAME] [-d DIRECTORY] [-flavor FLAVOR] [-lang [ c | cpp ]]
[-p PREFIX-MAP-FILE] [-P NAMESPACE=PREFIX]
[-n NAMESPACE-FILE] [-N NAMESPACE]
[-g NAMESPACE-FILE] [-G NAMESPACE] [-gg]
[-np] [-nb] [-nk] [-ns] [-nc] [-no-sources] [-nt] [-nf FUNCTION]
[-extra-cppflags CPPFLAGS] [-extra-ldflags LDFLAGS] [-extra-libs LIBS]
SCHEMA-FILENAME...
```

Description

The **globus-wsrf-cgen** tool generates C-language bindings from WSDL and XML Schema files. The input *SCHEMA-FILENAME* value should be either a WSDL document containing a service description or an XML schema file containing type definitions.

If a WSDL Schema file is specified as input, **globus-wsrf-cgen** generates a GPT source package containing client stubs, service skeleton and stubs, and type bindings for included schema types. If an XML Schema file is specified as input, it generates a GPT source package containing type bindings. A full description of the generated files is part of the [WSDL to C mapping document](#).

The full set of command-line options to **globus-wsrf-cgen** are:

| | |
|----------------------------|--|
| -help | Display a help message and exit |
| -dr | Dry-run: parse the command-line options and display the command-line arguments to the globus-wsdl-parser program. |
| -s <i>PACKAGE-NAME</i> | Use <i>PACKAGE-NAME</i> _bindings as the name for the generated GPT package |
| -sn <i>SERVICE-NAME</i> | Use <i>SERVICE-NAME</i> as the name of the service instead of the name in the WSDL schema document. |
| -d <i>DIRECTORY</i> | Generate the GPT source package in <i>DIRECTORY</i> , creating it if does not exist. |
| -flavor <i>FLAVOR</i> | Build the package using the <i>FLAVOR</i> GPT <i>flavor</i> |
| -lang <i>LANG</i> | Create the service implementation file with the extension matching <i>LANG</i> , either "c" or "cpp". See the limitations section for more details. |
| -p <i>PREFIX-MAP-FILE</i> | Use the contents of <i>PREFIX-MAP-FILE</i> to define the set of strings to prepend to elements, attributes, and types in various XML namespaces. See the namespace handling section of this document for more details. |
| -P <i>NAMESPACE=PREFIX</i> | Prepend element, attribute, and type names in the XML namespace <i>NAMESPACE</i> with the string <i>PREFIX</i> . See the namespace handling section of this document for more details. |
| -n <i>NAMESPACE-FILE</i> | Generate bindings for schemas in the XML namespaces contained in the <i>NAMESPACE-FILE</i> . See the namespace handling section of this document for more details. |
| -N <i>NAMESPACE</i> | Generate bindings for schemas in the XML namespace <i>NAMESPACE</i> . See the namespace handling section of this document for more details. |

| | |
|---|---|
| <code>-g <i>NAMESPACE-FILE</i></code> | Do not generate bindings for schemas in the XML namespaces contained in the <i>NAMESPACE-FILE</i> . See the namespace handling section of this document for more details. |
| <code>-G <i>NAMESPACE</i></code> | Do not generate bindings for schemas in the XML namespace <i>NAMESPACE</i> . See the namespace handling section of this document for more details. |
| <code>-gg</code> | Do not generate bindings for core WSRF namespaces. (Used internally only) |
| <code>-np</code> | Do not generate a GPT package. Only create source files from the schemas. Implies <code>-nb</code> . |
| <code>-nb</code> | Do not attempt to run configure and make dist on the generated GPT source package. |
| <code>-nk</code> | Do not generate a skeleton service implementation. Used in Makefiles for packages that want to generate the types at build time, but already contain a full implementation of the service. |
| <code>-ns</code> | Do not generate service bindings and skeletons. Useful for creating types- or client-only packages. |
| <code>-nc</code> | Do not generate client bindings. Useful for creating types- or service-only packages. |
| <code>-nt</code> | Do not generate type bindings. Useful for creating separate service or client bindings that depend on a common types package. |
| <code>-no-sources</code> | Delay generating C source files until the package is built. By default the package Makefile contains a list of source files. This option delays the creation of the files and the list until build time. This can be used to avoid storing dynamic files in a version control system. |
| <code>-nf <i>FUNCTION</i></code> | Do not generate an implementation of <i>FUNCTION</i> . This is useful if extra semantic information is needed to serialize or deserialize a particular data type (for example, the <code>wsnt:TopicExpressionType</code> requires different processing based on the value of the <code>Dialect</code> |
| <code>-extra-cppflags <i>CPP-FLAGS</i></code> | Add <i>CPPFLAGS</i> to the preprocessor command-line for this package. |
| <code>-extra-ldflags <i>LDFLAGS</i></code> | Add <i>LDFLAGS</i> to the linker command-line for this package. |
| <code>-extra-libs <i>LIBS</i></code> | Add <i>LIBS</i> to the libraries to link with this package. |

Namespace Handling

XML and WSDL schemas generally contain a `targetNamespace` attribute which distinguishes operations, elements, attributes, type, etc from others with the same name. The C language does not define namespaces. **globus-wsrf-cgen** instead uses prefixes to distinguish similarly-named data types and functions. There are two ways to define a namespace prefix with **globus-wsrf-cgen**. The `-P` command-line option defines a single namespace prefix, and the `-p` command-line option instructs **globus-wsrf-cgen** to load a set of prefix definitions from a file (one per line).

For example, consider the namespace `http://counter.com` from the sample `CounterService`. In the schema for that service, there is an element named `Value`. the command-line option `-P http://counter.com=counter_` will cause **globus-wsrf-cgen** to generate bindings for that element with the name `counter_Value`.

If a service is built from several namespaces it might make sense instead to use the `-P` parameter instead. Using the same service as the previous example, we could instead create a file containing

```
http://counter.com=counter_
http://another.counter.com=another_counter_
```

to generate C prefixes for multiple namespaces.

A service may be composed of operations and data types from multiple namespaces. By default **globus-wsrf-cgen** generates bindings for all namespaces except those used by the core WSRF specifications. These are (along with their C prefixes):

Table 37. WSRF Core Namespaces and C Prefixes

| | |
|--|--------|
| http://www.w3.org/XML/1998/namespace | xml_ |
| http://www.w3.org/2001/XMLSchema | xsd_ |
| http://www.w3.org/2005/08/addressing | wsa_ |
| http://docs.oasis-open.org/wsrf/r-2 | wsr_ |
| http://docs.oasis-open.org/wsrf/rw-2 | wsrw_ |
| http://docs.oasis-open.org/wsrf/bf-2 | wsbf_ |
| http://docs.oasis-open.org/wsrf/rp-2 | wsrp_ |
| http://docs.oasis-open.org/wsrf/rpw-2 | wsrpw_ |
| http://docs.oasis-open.org/wsrf/rl-2 | wsrl_ |
| http://docs.oasis-open.org/wsrf/rlw-2 | wsrlw_ |
| http://docs.oasis-open.org/wsrf/sg-2 | wssg_ |
| http://docs.oasis-open.org/wsrf/sgw-2 | wssgw_ |
| http://docs.oasis-open.org/wsn/b-2 | wsnt_ |
| http://docs.oasis-open.org/wsn/bw-2 | wsntw_ |
| http://docs.oasis-open.org/wsn/t-1 | wstop_ |
| http://schemas.xmlsoap.org/ws/2002/12/policy | wsp_ |
| http://schemas.xmlsoap.org/ws/2002/07/utility | wsu_ |
| http://schemas.xmlsoap.org/ws/2004/04/trust | wst_ |
| http://www.w3.org/2000/09/xmlsig# | ds_ |
| http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd | wsse_ |
| http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd | wsseu_ |
| http://schemas.xmlsoap.org/ws/2004/04/sc | wsc_ |
| http://schemas.xmlsoap.org/ws/2004/09/enumeration | wsen_ |

Often it is enough for a package to contain bindings for the namespaces unique to the service and compile against other packages which contain the bindings for the other namespaces. This control can be done positively via the `-N` and `-n` command-line options.

For example, to generate bindings for the `http://counter.com` namespace *only*, pass the command-line option `-N http://counter.com`. To generate for both the `http://counter.com` and `http://another.counter.com` namespaces, either pass multiple `-N` options with one namespace each, or create a file containing:

```
http://counter.com
http://another.counter.com
```

and pass the name of the file to `globus-wsrf-cgen` as the parameter to the `-n` command-line option.

Examples

Here is a brief example of the **globus-wsrf-cgen** command. For more details, see the [tutorials in the C WS Core developer documentation](#).

Create bindings for a service in the http://counter.com namespace:

```
% globus-wsrf-cgen -d counter \  
  -N http://counter.com \  
  -s counter \  
  -P http://counter.com=counter_ \  
  $GLOBUS_LOCATION/share/schemas/core/samples/counter_service.wsdl
```

Creating Bindings Package

A new package has been created at /home/griduser/counter/counter_bindings-1.2.tar.gz
To install, use the following command:

```
$GLOBUS_LOCATION/sbin/gpt-build /Users/bester/tmp/foo/counter/counter_bindings-1.2.tar.gz  
%
```

Limitations

- This program only generates bindings from document/literal style WSDL schemas. IBM developerworks has [an article describing the different WSDL schema styles](#)¹.
- The bindings generated when `-lang cpp` is used are ANSI-C. However, all C++ keywords are avoided and no constructs that differ between C and C++ are used. This command-line option merely creates a makefile which compiles the service implementation with the C++ compiler.
- Not all XML Schema constructs are supported. In particular, abstract types, substitution groups, and nested sequences are not implemented.

¹ <http://www.ibm.com/developerworks/webservices/library/ws-whichwsdl/>

Name

globus-wsrf-destroy -- Set the scheduled termination time for a WSRF resource.

```
globus-wsrf-destroy [OPTIONS]... SERVICE-SPECIFIER
```

Tool description

Set the scheduled termination time for a WSRF resource.

Command syntax

```
globus-wsrf-destroy [OPTIONS]... SERVICE-SPECIFIER
```

Table 38. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart , while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME. The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME. |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

Examples:

```
% globus-wsrf-destroy -e widget.epr  
Resource destroyed
```

Contents of *widget.epr*:

```
<ns01:EndpointReference xmlns:ns01="http://schemas.xmlsoap.org/ws/2004/03/addressing">  
  <ns01:Address>http://globus.my.org:8080/wsrf/services/WidgetService</ns01:Address>  
  <ns01:ReferenceProperties>  
    <ResourceID xmlns:ns02="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns03="http://"  
  </ns01:ReferenceProperties>  
</ns01:EndpointReference>
```

Output and Exit Code

If the resource is destroyed successfully, the string `Resource destroyed` will be displayed to *stdout* and the program will terminate with exit code 0. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsrf-set-termination-time -- Set the scheduled termination time for a WSRF resource.

```
globus-wsrf-set-termination-time [OPTIONS]... SERVICE-SPECIFIER TERMINATION-TIME
```

Tool description

Set the scheduled termination time for a WSRF resource.

Command syntax

```
globus-wsrf-set-termination-time [OPTIONS]... SERVICE-SPECIFIER TERMINATION-TIME
```

Table 39. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart . while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME . The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME . |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

TERMINATION-TERMINATION: [SECONDS | 'infinity']

Examples:

```
% globus-wsrf-set-termination-time -e widget.epr `expr 24 \* 60 \* 60`  
Termination time set to 2006-05-31T20:18:43Z
```

Contents of *widget.epr*:

```
<ns01:EndpointReference xmlns:ns01="http://schemas.xmlsoap.org/ws/2004/03/addressing">  
  <ns01:Address>http://globus.my.org:8080/wsrf/services/WidgetService</ns01:Address>  
  <ns01:ReferenceProperties>  
    <ResourceID xmlns:ns02="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns03="http://"  
  </ns01:ReferenceProperties>  
</ns01:EndpointReference>
```

Output and Exit Code

If the termination time is set successfully, the string Termination time set to YYYY-MM-DD-THH:MM:SS[.MSEC]Z will be displayed to *stdout* and the program will terminate with exit code 0. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsrf-query -- Query a WSRF resource's Resource Property document

globus-wsrf-query [OPTIONS]... SERVICE-SPECIFIER QUERY-EXPRESSION

Tool description

Perform an XPATH query on a resource property document.

Command syntax

globus-wsrf-query [OPTIONS]... SERVICE-SPECIFIER QUERY-EXPRESSION

Table 40. Application-specific options

| | |
|--|--|
| -n ---nsMapFile FILENAME. | Use the namespace map entries in <i>FILENAME</i> in the XPATH context. |
| -N --namespace PREFIX=NAMESPACE-URI | Create a namespace mapping in the XPATH context for the <i>PREFIX</i> string to resolve to the <i>NAMESPACE-URI</i> namespace. |
| -D --dialect DIALECT-URI | Set query dialect to <i>DIALECT-URI</i> . The value targeted will be interpreted as http://wsrf.globus.org/core/query/targetedXPath (default: http://www.w3.org/TR/1999/REC-xpath-19991116). |

Table 41. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart . while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME . The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME . |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

QUERY-EXPRESSION: XPath-Expression-String

Examples:

```
% globus-wsrf-query -e widget.epr "//*[local-name() = 'CurrentTime']"
<ns0:CurrentTime
  xmlns:ns0="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns1="http://www.w3.org/2001/XMLSchema"
  xmlns:ns2="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft
  ns0:type="ns1:dateTime">2006-05-30T13:53:15Z</ns0:CurrentTime>
```

```
% globus-wsrf-query -e widget.epr "//*[local-name() = 'CurrentTime']/text()"
2006-05-30T13:53:35Z
```

```
% globus-wsrf-query -e widget.epr \
    -N wsrl=http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-
    "/wsrl:CurrentTime/text()"
2006-05-30T13:54:36Z
```

Contents of *widget.epr*:

```
<ns01:EndpointReference xmlns:ns01="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns01:Address>http://globus.my.org:8080/wsrf/services/WidgetService</ns01:Address>
  <ns01:ReferenceProperties>
    <ResourceID xmlns:ns02="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns03="http://
  </ns01:ReferenceProperties>
</ns01:EndpointReference>
```

Limitations

- The namespace mapping option and use of namespace prefixes in the *XPath-Expression-String* does not work when communicating with the Java container unless the *http://wsrf.globus.org/core/query/targetedXPath* dialect is used.

Output and Exit Code

If the query is successful, the program displays the output of the query to *stdout* and terminates with exit code 0. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsrf-get-property -- Get a resource property's value

globus-wsrf-get-property [OPTIONS]... SERVICE-SPECIFIER PROPERTY-NAME

Tool description

Get the value of a resource property from a WSRF resource.

Command syntax

globus-wsrf-get-property [OPTIONS]... SERVICE-SPECIFIER PROPERTY-NAME

Table 42. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart , while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME. The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME. |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

PROPERTY-NAME: [{Namespace-URI}]Property-Name

Example:

```
% globus-wsrf-get-property -e widget.epr \  
    '{http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.xsd  
  
<ns02:CurrentTime  
    xmlns:ns00="http://www.w3.org/2001/XMLSchema-instance"  
    xmlns:ns01="http://www.w3.org/2001/XMLSchema"  
    xmlns:ns02="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft  
    ns00:type="ns01:dateTime">2006-05-30T14:26:35Z</ns02:CurrentTime>
```

Output and Exit Code

If the property exists, its values (if any) are displayed to *stdout* and the program terminates with exit code 0. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsrf-get-properties -- Get multiple resource property value

globus-wsrf-get-properties [OPTIONS]... SERVICE-SPECIFIER PROPERTY-NAME...

Tool description

Get the value of multiple resource properties from a WSRF resource.

Command syntax

globus-wsrf-get-properties [OPTIONS]... SERVICE-SPECIFIER PROPERTY-NAME...

Table 43. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart . while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME . The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME . |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

PROPERTY-NAME: [{Namespace-URI}]Property-Name

Example:

```
% globus-wsrf-get-properties \
  -s http://grid.example.org:8080/wsrf/services/WidgetService \
  -k "{http://www.globus.org/namespaces/2004/06/core}WidgetKey" 123 \
  "{http://widgets.com}foo" \
  "{http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.xsd}foo"
<ns02:foo
  xmlns:ns0="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns1="http://www.w3.org/2001/XMLSchema"
  xmlns:ns2="http://widgets.com"
  ns0:type="ns01:string">
Foo Value String
</ns02:foo><ns03:CurrentTime
  xmlns:ns0="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns1="http://www.w3.org/2001/XMLSchema"
  xmlns:ns3="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.xsd"
  ns0:type="ns01:dateTime">2006-05-30T16:04:15Z</ns03:CurrentTime>
```

Output and Exit Code

If the properties exist, their values (if any) are displayed to *stdout* and the program terminates with exit code 0. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsrf-insert-property -- Insert a resource property value

```
globus-wsrf-insert-property [OPTIONS]... SERVICE-SPECIFIER PROPERTY-VALUE-FILE-NAME
```

Tool description

Insert a resource property into a WSRF resource's Resource Properties document. The new property will be read from the XML file specified by *PROPERTY-VALUE-FILENAME*.

Command syntax

```
globus-wsrf-insert-property [OPTIONS]... SERVICE-SPECIFIER PROPERTY-VALUE-FILENAME...
```

Table 44. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart . while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME . The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME . |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

Example:

```
% globus-wsrf-insert-property -e widget.epr widget:foo.xml
```

Contents of *widget.epr*:

```
<ns01:EndpointReference xmlns:ns01="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns01:Address>http://globus.my.org:8080/wsrf/services/WidgetService</ns01:Address>
  <ns01:ReferenceProperties>
    <ResourceID xmlns:ns02="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns03="http://
  </ns01:ReferenceProperties>
```

```
</ns01:EndpointReference>
```

Contents of *widget:foo.xml*:

```
<doc>
  <foo xmlns="http://widgets.com"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xsd:string">
    Foo Value String
  </foo>
</doc>
```

Output and Exit Code

If the property is inserted successfully, the program terminates with exit code 0. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsrf-update-property -- Update a resource property value

```
globus-wsrf-update-property [OPTIONS]... SERVICE-SPECIFIER PROPERTY-VALUE-FILE-NAME
```

Tool description

Update a resource property in a WSRF resource's Resource Properties document. The property's new value will be read from the XML file specified by *PROPERTY-VALUE-FILENAME*. An update operation will replace the value(s) of the resource property with the new value(s) in the property file.

Command syntax

```
globus-wsrf-update-property [OPTIONS]... SERVICE-SPECIFIER PROPERTY-VALUE-FILENAME
```

Table 45. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart . while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME . The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME . |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

Example:

```
% globus-wsrf-update-property -e widget.epr widget:foo.xml
```

Contents of *widget.epr*:

```
<ns01:EndpointReference xmlns:ns01="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns01:Address>http://globus.my.org:8080/wsrf/services/WidgetService</ns01:Address>
  <ns01:ReferenceProperties>
    <ResourceID xmlns:ns02="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns03="http://
  </ns01:ReferenceProperties>
</ns01:EndpointReference>
```

Contents of *widget:foo.xml*:

```
<doc>
  <foo xmlns="http://widgets.com"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xsd:string">
    Foo Value String
  </foo>
</doc>
```

Output and Exit Code

If the property update is successful without any output, then the program terminates with exit code 0. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsrf-delete-property -- Delete a resource property

globus-wsrf-delete-property [OPTIONS] SERVICE-SPECIFIER PROPERTY-NAME

Tool description

Delete a resource property from a WSRF resource.

Command syntax

globus-wsrf-delete-property [OPTIONS]... SERVICE-SPECIFIER PROPERTY-NAME

Table 46. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart . while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME. The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME. |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

PROPERTY-NAME: [{Namespace-URI}]Property-Name

Example:

```
% globus-wsrf-delete-property \  
-s http://grid.example.org:8080/wsrf/services/WidgetService \  
-k "{http://www.globus.org/namespaces/2004/06/core}WidgetKey" 123 \  
"{http://widgets.com}foo"
```

Output and Exit Code

If the property is successfully deleted, **globus-wsrf-delete-property** will not print out any output and will terminate with the exit code 0. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsn-get-current-message -- Get the current message associated with a specified topic

globus-wsn-get-current-message [OPTIONS] SERVICE-SPECIFIER TOPIC-EXPRESSION

Tool description

Get the current message associated with a specified topic.

Command syntax

globus-wsn-get-current-message [OPTIONS]... SERVICE-SPECIFIER TOPIC-EXPRESSION

Table 47. Application-specific options

| | |
|--|--|
| -N --namespace PREFIX=NAMESPACE-URI | Create a namespace mapping in the XPATH context for the <i>PREFIX</i> string to resolve to the <i>NAMESPACE-URI</i> namespace in the Topic Expression. |
| -D --dialect DIALECT-URI | Set the Topic Expression dialect to <i>DIALECT-URI</i> . If not specified, the dialect is chosen automatically between <i>http://docs.oasis-open.org/wsn/2004/06/TopicExpression/Simple</i> , <i>http://docs.oasis-open.org/wsn/2004/06/TopicExpression/Concrete</i> , and <i>http://docs.oasis-open.org/wsn/2004/06/TopicExpression/Full</i> based on the presence of substrings '*', '/', ' ', and '.' in the Topic Expression string. |

Table 48. Common options

| | |
|--|---|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart. while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME. The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME. |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

TOPIC-EXPRESSION: [{Namespace-URI} | prefix ':']RootTopic[/ChildTopic]...
 TOPIC-EXPRESSION ['|' TOPIC-EXPRESSION]
 RootChild or ChildTopic may contain '*' (wildcard) and/or
 '/' (all descendents)

Example:

```
% globus-wsn-get-current-message \
  -e widget.epr \
  -N wsrl=http://docs.oasis-open.org/wsrfl/2004/06/wsrfl-WS-ResourceLifetime-1.2-draft-
  'wsrl:TerminationTime'
```

```
<ns0:ResourcePropertyValueChangeNotification
  xmlns:ns0="http://docs.oasis-open.org/wsrfl/2004/06/wsrfl-WS-ResourceProperties-1.2-draft
```

```
xmlns:ns01="http://www.w3.org/2001/XMLSchema-instance"
ns01:type="ns00:ResourcePropertyValueChangeNotificationType">
  <ns00:NewValue
    ns01:type="ns00:NewValueType">
      <ns03:TerminationTime
        xmlns:ns02="http://www.w3.org/2001/XMLSchema"
        xmlns:ns03="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2"
        ns01:type="ns02:dateTime">2006-05-31T20:10:08Z</ns03:TerminationTime>
      </ns00:NewValue>
    </ns00:ResourcePropertyValueChangeNotification>
```

Contents of *widget.epr*:

```
<ns01:EndpointReference xmlns:ns01="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns01:Address>http://globus.my.org:8080/wsrf/services/WidgetService</ns01:Address>
  <ns01:ReferenceProperties>
    <ResourceID
      xmlns:ns02="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ns03="http://www.w3.org/2001/XMLSchema"
      ns02:type="ns03:string">7f554f7c-efd9-11da-97a5-00096b86f788</ResourceID>
    </ns01:ReferenceProperties>
  </ns01:EndpointReference>
```

Output and Exit Code

If the Topic exists and has a current message, **globus-wsn-get-current-message** will print the current message value to *stdout* and then terminate with the exit code 0. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsn-pause-subscription -- Pause a WSRF notification subscription.

globus-wsn-pause-subscription [OPTIONS] SERVICE-SPECIFIER

Tool description

Pause a WSRF notification subscription.

Command syntax

globus-wsn-pause-subscription [OPTIONS]... SERVICE-SPECIFIER TOPIC-EXPRESSION

Table 49. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart , while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME. The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME. |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

Example:

```
% globus-wsn-pause-subscription \  
    -e subscription.epr
```

Contents of *subscription.epr*:

```
<ns00:EndpointReference  
    xmlns:ns00="http://schemas.xmlsoap.org/ws/2004/03/addressing">  
  <ns00:Address>http://globus.my.org:8080/wsrfl/services/SubscriptionManagerService</ns00:Address>  
  <ns00:ReferenceProperties>  
    <ns03:ResourceID  
      xmlns:ns01="http://www.w3.org/2001/XMLSchema-instance"  
      xmlns:ns02="http://www.w3.org/2001/XMLSchema"  
      xmlns:ns03="http://www.globus.org/docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification"  
      ns01:type="ns02:string">7d6430e4-f019-11da-1b9-00096b86f788</ns03:ResourceID>  
    </ns00:ReferenceProperties>  
  </ns00:EndpointReference>
```

Output and Exit Code

If the subscription is successfully paused, **globus-wsn-pause-subscription** will terminate with the exit code 0. No further notifications should be expected on the Subscription resource until it is resumed again. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsn-resume-subscription -- Resume a WSRF notification subscription.

globus-wsn-resume-subscription [OPTIONS] SERVICE-SPECIFIER

Tool description

Resume a subscription.

Command syntax

globus-wsn-resume-subscription [OPTIONS]... SERVICE-SPECIFIER TOPIC-EXPRESSION

Table 50. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart . while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME . The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME . |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

Example:

```
% globus-wsn-resume-subscription \
    -e subscription.epr
```

Contents of *subscription.epr*:

```
<ns00:EndpointReference
  xmlns:ns00="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns00:Address>http://globus.my.org:8080/wsrfl/services/SubscriptionManagerService</ns00:Address>
  <ns00:ReferenceProperties>
    <ns03:ResourceID
      xmlns:ns01="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ns02="http://www.w3.org/2001/XMLSchema"
      xmlns:ns03="http://www.globus.org/docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification"
      ns01:type="ns02:string">7d6430e4-f019-11da-afb9-00096b86f788</ns03:ResourceID>
    </ns00:ReferenceProperties>
  </ns00:EndpointReference>
```

Output and Exit Code

If the subscription is successfully resumed, **globus-wsn-resume-subscription** will terminate with the exit code 0. Notifications should again flow to the Subscription resource. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

Name

globus-wsn-subscribe -- Subscribe for notification for a specified topic.

globus-wsn-subscribe [OPTIONS] SERVICE-SPECIFIER TOPIC-EXPRESSION

Tool description

Subscribe for notification for a specified topic.

Command syntax

globus-wsn-subscribe [OPTIONS]... SERVICE-SPECIFIER TOPIC-EXPRESSION

Table 51. Application-specific options

| | |
|--|---|
| -b --subEpr FILENAME | Save the Subscription Manager EPR in <i>FILENAME</i> . This EPR file can be used with the <code>globus-wsn-pause-subscription</code> and <code>globus-wsn-resume-subscription</code> commands |
| -N --namespace PREFIX=NAMESPACE-URI | Create a namespace mapping in the XPATH context for the <i>PREFIX</i> string to resolve to the <i>NAMESPACE-URI</i> namespace in the Topic Expression. |
| -D --dialect DIALECT-URI | Set the Topic Expression dialect to <i>DIALECT-URI</i> . If not specified, the dialect is chosen automatically between <code>http://docs.oasis-open.org/wsn/2004/06/TopicExpression/Simple</code> , <code>http://docs.oasis-open.org/wsn/2004/06/TopicExpression/Concrete</code> , and <code>http://docs.oasis-open.org/wsn/2004/06/TopicExpression/Full</code> based on the presence of substrings <code>*</code> , <code>/</code> , <code> </code> , and <code>'</code> in the Topic Expression string. |

Table 52. Common options

| | |
|--|---|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart . while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME . The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME . |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

SERVICE-SPECIFIER: [-s URI [-k KEY VALUE] | -e FILENAME]

TOPIC-EXPRESSION: [{Namespace-URI} | prefix ':']RootTopic[/ChildTopic]...
 TOPIC-EXPRESSION ['|' TOPIC-EXPRESSION]
 RootChild or ChildTopic may contain '*' (wildcard) and/or
 '/' (all descendents)

Example:

```
% globus-wsn-subscribe \
    -e counter.epr \
    -N counter=http://www.counter.com \
    'counter:Value'
<ns02:Value
  xmlns:ns00="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns01="http://www.w3.org/2001/XMLSchema"
```

```

    xmlns:ns02="http://counter.com" ns00:type="ns01:int">10</ns02:Value>
<ns02:Value
  xmlns:ns00="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns01="http://www.w3.org/2001/XMLSchema"
  xmlns:ns02="http://counter.com"
  ns00:type="ns01:int">20</ns02:Value>

```

Contents of *counter.epr*:

```

<ns01:EndpointReference
  xmlns:ns01="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns01:Address>http://globus.my.org:8080/wsrf/services/CounterService</ns01:Address>
  <ns01:ReferenceProperties>
    <ns04:CounterKey
      xmlns:ns02="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ns03="http://www.w3.org/2001/XMLSchema"
      xmlns:ns04="http://counter.com/service"
      ns02:type="ns03:string">1804289383</ns04:CounterKey>
    </ns01:ReferenceProperties>
  </ns01:EndpointReference>

```

Output and Exit Code

globus-wsn-subscribe will print the contents of notification message to *stdout*. If the message is a ResourcePropertyValueChangedNotification message, then only the NewValue subelement will be displayed. Otherwise, the entire message will be displayed. This program will run until terminated by a signal. In the case of an error, the type of error will be displayed to *stderr* and the program will terminate with a non-0 exit code.

GSI Commands

Name

grid-cert-diagnostics -- Print diagnostic information about certificates and keys

grid-cert-diagnostics [-h] [-p]

Description

The **grid-cert-diagnostics** command displays information about the current user's security environment, including information about security-related environment variables, security directory search path, personal key and certificates, and trusted certificates. It is intended to provide information to help diagnose problems using GSI security.

The full set of command-line options to **grid-cert-diagnostics** consists of:

| | |
|----|---|
| -h | Display a help message and exit |
| -p | Display information about the personal certificate and key that is the current user's default credential. |

Examples

In this example, we see the default mode of checking the default security environment for the system, without processing the user's key and certificate. Note the user receives a warning about a `cog.properties` and about an expired CA certificate.

```
% grid-cert-diagnostics
```

```
Checking Environment Variables
```

```
=====
```

```
Checking if X509_CERT_DIR is set... no  
Checking if X509_USER_CERT is set... no  
Checking if X509_USER_KEY is set... no  
Checking if X509_USER_PROXY is set... no
```

```
Checking Security Directories
```

```
=====
```

```
Determining trusted cert path... /etc/grid-security/certificates  
Checking for cog.properties... found  
    WARNING: If the cog.properties file contains security properties,  
             Java apps will ignore the security paths described in the GSI  
             documentation
```

```
Checking trusted certificates...
```

```
=====
```

```
Getting trusted certificate list...
```

```
Checking CA file /etc/grid-security/certificates/1c4f4c48.0... ok  
Verifying certificate chain for "/etc/grid-security/certificates/1c3f2ca8.0"... ok  
Checking CA file /etc/grid-security/certificates/9d8788eb.0... ok  
Verifying certificate chain for "/etc/grid-security/certificates/9d8753eb.0"... failed  
    globus_credential: Error verifying credential: Failed to verify credential  
    globus_gsi_callback_module: Could not verify credential  
    globus_gsi_callback_module: The certificate has expired:
```

Credential with subject: /DC=org/DC=example/OU=grid/CN=CA has expired.

In this example, we show a user with a mismatched private key and certificate:

```
% grid-cert-diagnostics -p
```

```
Checking Environment Variables
```

```
=====
```

```
Checking if X509_CERT_DIR is set... no  
Checking if X509_USER_CERT is set... no  
Checking if X509_USER_KEY is set... no  
Checking if X509_USER_PROXY is set... no
```

```
Checking Security Directories
```

```
=====
```

```
Determining trusted cert path... /etc/grid-security/certificates  
Checking for cog.properties... not found
```

```
Checking Default Credentials
```

```
=====
```

```
Determining certificate and key file names... ok  
Certificate Path: "/home/juser/.globus/usercert.pem"  
Key Path: "/home/juser/.globus/userkey.pem"  
Reading certificate... ok  
Reading private key...  
ok
```

```
Checking Certificate Subject...
```

```
"/O=Grid/OU=Example/OU=User/CN=Joe User"
```

```
Checking cert... ok
```

```
Checking key... ok
```

```
Checking that certificate contains an RSA key... ok
```

```
Checking that private key is an RSA key... ok
```

```
Checking that public and private keys have the same modulus... failed
```

```
Private key modulus: D294849E37F048C3B5ACEEF2CCDF97D88B679C361E29D5CB5  
219C3E948F3E530CFC609489759E1D751F0ACFF0515A614276A0F4C11A57D92D7165B8  
FA64E3140155DE448D45C182F4657DA13EDA288423F5B9D169DFF3822EFD81EB2E6403  
CE3CB4CCF96B65284D92592BB1673A18354DA241B9AFD7F494E54F63A93E15DCAE2  
Public key modulus : C002C7B329B13BFA87BAF214EACE3DC3D490165ACEB791790  
600708C544175D9193C9BAC5AED03B7CB49BB6AE6D29B7E635FAC751E9A6D1CEA98022  
6F1B63002902D6623A319E4682E7BFB0968DCE962CF218AAD95FAAD6A0BA5C42AA9AAF  
7FDD32B37C6E2B2FF0E311310AA55FFB9EAFDF5B995C7D9EEAD8D5D81F3531E0AE5
```

```
Certificate and and private key don't match
```

Name

grid-cert-info -- Display certificate information

```
grid-cert-info [-help] [-version]
[-file CERTIFICATE-FILENAME]
[-all] [-subject] [-issuer] [-issuerhash] [-startdate] [-enddate]
```

Description

The **grid-cert-info** displays information from a user's credential, or from any X.509 certificate if the `-file CERTIFICATE-FILENAME` is used. By default, a text representation of the entire certificate is displayed. If more than one display option is present on the command line, the output is generated in the order the options occur on the command line.

The following search order is used to locate the default certificate:

- `$X509_USER_CERT`
- `$HOME/.globus/usercert.pem`
- `$HOME/.globus/usercred.p12`

If the certificate is encoded in pkcs12, **grid-cert-info** will prompt for the password used to protect the `.p12` file.

The full set of command-line options to **grid-cert-info** is:

| | |
|---|--|
| <code>-help</code> | Print help information and exit |
| <code>-version</code> | Print version information and exit |
| <code>-file CERTIFICATE-FILENAME</code> | Read credential from <code>CERTIFICATE-FILENAME</code> instead of the default location. The file must have a <code>.pem</code> or <code>.p12</code> extension. |
| <code>-all</code> | Print all information from the certificate. This is the default unless any of the following options are given. |
| <code>-subject</code> | Print the subject name of the certificate. |
| <code>-issuer</code> | Print the subject name of the issuer of the certificate. This is the subject name of the <i>Certificate Authority</i> which signed the certificate. |
| <code>-issuerhash</code> | Print the hash of the name of the issuer of the certificate. This is the hash of the Certificate Authority which signed the certificate. |
| <code>-startdate</code> | Print the date and time from which the certificate is valid |
| <code>-enddate</code> | Print the date and time when the certificate expires. |

Examples

Print out the date range when a certificate is valid:

```
% grid-cert-info -startdate -enddate
```

```
Oct 29 13:09:42 2007 GMT
Oct 28 13:09:42 2008 GMT
```

Note that in this example, the start date is printed first, based on the order of the command-line options.

Limitations

The `-issuerhash` fails with some versions of OpenSSL.

Name

grid-cert-request -- Create a certificate request

```
grid-cert-request [-help] [-version] [-verbose] [-force]
[-commonname NAME] [-service SERVICE] [-host FQDN] [-interactive]
[-dir DIRECTORY] [-prefix PREFIX] [-ca [HASH]] [-nopw]
```

Description

grid-cert-request generates a public/private key pair and an X.509 certificate request containing the public key and a subject name. By default, it generates a request for a user certificate for the invoking user. **grid-cert-request** can also be used to create host or service certificates based on command-line options. At least one *Certificate Authority* must be configured to use with the Globus Toolkit in order for this command to succeed.

Complete set of options to **grid-cert-request** is:

| | |
|-------------------------|---|
| -help | Print help information and exit |
| -version | Print version information and exit |
| -verbose | Don't clear screen after running OpenSSL |
| -force | Overwrite an existing certificate request if present. |
| -commonname <i>NAME</i> | Construct a subject name with <i>NAME</i> as the final name component. By default, the subject name is inferred from the output of the finger program. If that fails, grid-cert-request will prompt for a name. |
| -service <i>SERVICE</i> | Construct a subject name with the common name constructed from the <i>SERVICE</i> name and the hostname joined by the / character. The <i>-service</i> requires that the <i>-host</i> option also be used. The private key created for a service certificate request is not encrypted. |
| -host <i>FQDN</i> | Construct a subject name with <i>FQDN</i> as the name of the host. This must be a fully-qualified name in dotted string notation (e.g. <i>grid.example.org</i>). If no service is specified by the <i>-service</i> option, the subject name will be <i>host/FQDN</i> . The private key created for a host certificate request is not encrypted. By default the host certificate request and key are created in <i>/etc/grid-security</i> . |
| -interactive | Interactively prompt for the components of the certificate subject name. |
| -dir <i>DIRECTORY</i> | Write the certificate request and key to <i>DIRECTORY</i> , creating it if the directory does not exist. By default, the certificate request and key are placed in <i>\$HOME/.globus</i> |
| -prefix <i>PREFIX</i> | Prepend the string <i>PREFIX</i> to the certificate, key, and request filenames. The default prefix is <i>user</i> for user certificates and <i>host</i> for host certificates. |
| -ca <i>HASH</i> | Choose a non-default Certificate Authority configuration to construct the certificate request. If <i>HASH</i> is present on the command line, then grid-cert-request will use that certificate authority's configuration. Otherwise, it will prompt the user for a CA to choose from the list of configured CAs. |
| -nopw | Create a private key without a password. This may be a security risk if the file permissions of the private key are not carefully maintained. |

Examples

Request a user certificate:

```
% grid-cert-request
```

```
A certificate request and private key is being created.
You will be asked to enter a PEM pass phrase.
This pass phrase is akin to your account password,
and is used to protect your key file.
If you forget your pass phrase, you will need to
obtain a new certificate.
```

```
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to '/home/juser/.globus/userkey.pem'
Enter PEM pass phrase:
```

A private key and a certificate request has been generated with the subject:

```
/O=Grid/OU=Example/OU=User/CN=Joe User
```

If the CN=Joe User is not appropriate, rerun this script with the `-force -cn "Common Name"` options.

```
Your private key is stored in /home/juser/.globus/userkey.pem
Your request is stored in /home/juser/.globus/usercert_request.pem
```

Please e-mail the request to the Globus Certificate Service `ca@grid.example.org`
You may use a command similar to the following:

```
cat /home/juser/.globus/usercert_request.pem | mail ca@grid.example.org
```

Only use the above if this machine can send AND receive e-mail. if not, please mail using some other method.

Your certificate will be mailed to you within two working days.
If you receive no response, contact Globus Certificate Service at `ca@grid.example.org`

Request a host certificate, putting the request and key files in the `$HOME/.globus/host` directory.

```
% grid-cert-request -host grid.example.org -dir $HOME/.globus/host
```

A private host key and a certificate request has been generated with the subject:

```
/O=Grid/OU=Example/OU=User/CN=host/grid.example.org
```

The private key is stored in /tmp/examplegrid/hostkey.pem
The request is stored in /tmp/examplegrid/hostcert_request.pem

Please e-mail the request to the Globus Certificate Service ca@grid.example.org
You may use a command similar to the following:

```
cat /tmp/examplegrid/hostcert_request.pem | mail ca@grid.example.org
```

Only use the above if this machine can send AND receive e-mail. if not, please mail using some other method.

Your certificate will be mailed to you within two working days.
If you receive no response, contact Globus Certificate Service at ca@grid.example.org

Limitations

Only supports PEM-encoded keys, certificates and certificate requests.

Name

grid-default-ca -- Set the default CA to use for certificate requests

grid-default-ca [-help] [-list] [-ca *CA-HASH*] [-dir *SECURITY-DIRECTORY*]

Description

The **grid-default-ca** program sets the default CA used by **grid-cert-request**. Based on the default CA choice, **grid-cert-request** will create a certificate request that matches the CA's naming policies.

If the `-ca` option is not provided on the command-line, **grid-default-ca** will display a list of available Certificate Authorities and prompt the user to choose one.

The full set of command-line options to **grid-default-ca** are:

| | |
|---|--|
| <code>-help</code> | Display a help message and exit |
| <code>-list</code> | List the available CAs but do not alter the default |
| <code>-ca <i>CA-HASH</i></code> | Select the default CA whose subject name hash matches <i>CA-HASH</i> . |
| <code>-dir <i>SECURITY-DIRECTORY</i></code> | Search <i>SECURITY-DIRECTORY</i> for additional CA certificates. |

Examples

Show what certificate authorities are in the trusted cert directory:

```
% grid-default-ca -list
```

The available CA configurations installed on this host are:

Directory: /etc/grid-security/certificates

- 1) 1c3f2ca8 - /DC=org/DC=DOEGrids/OU=Certificate Authorities/CN=DOEGrids CA 1
- 2) 3d8e6ce8 - /O=Grid/CN=Example CA
- 3) 6349a761 - /O=DOE Science Grid/OU=Certificate Authorities/CN=Certificate Manager
- 4) b38b4d8c - /C=US/O=Globus Alliance/CN=Globus Certificate Service

The default CA is: /C=US/O=Globus Alliance/CN=Globus Certificate Service
Location: /etc/grid-security/certificates/b38b4d8c.0

Change the default CA to be *DOEGrids CA 1*:

```
% grid-default-ca
```

The available CA configurations installed on this host are:

Directory: /etc/grid-security/certificates

- 1) 1c3f2ca8 - /DC=org/DC=DOEGrids/OU=Certificate Authorities/CN=DOEGrids CA 1
- 2) 3d8e6ce8 - /O=Grid/CN=Example CA
- 3) 6349a761 - /O=DOE Science Grid/OU=Certificate Authorities/CN=Certificate Manager
- 4) b38b4d8c - /C=US/O=Globus Alliance/CN=Globus Certificate Service

The default CA is: /C=US/O=Globus Alliance/CN=Globus Certificate Service
Location: /etc/grid-security/certificates/b38b4d8c.0

Enter the index number of the CA to set as the default [q to quit]: 1

setting the default CA to: /DC=org/DC=DOEGrids/OU=Certificate Authorities/CN=DOEGrids CA 1

linking /etc/grid-security/certificates/grid-security.conf.1c3f2ca8 to
/etc/grid-security/grid-security.conf

linking /etc/grid-security/certificates/globus-host-ssl.conf.1c3f2ca8 to
/etc/grid-security/globus-host-ssl.conf

linking /etc/grid-security/certificates/globus-user-ssl.conf.1c3f2ca8 to
/etc/grid-security/globus-user-ssl.conf

...done.

Limitations

Displays all CAs in the output, even those where the globus-user-ssl.conf and globus-host-ssl.conf files are not installed in the trusted certificate directory. If one of those is chosen, **grid-default-ca** displays an error and exits.

Name

grid-change-pass-phrase -- Change the pass phrase on a private key

grid-change-pass-phrase

Tool description

grid-change-pass-phrase allows one to change the passphrase that protects the private key.

Command syntax

```
grid-change-pass-phrase [-help] [-version] [-file private_key_file]
```

Changes the passphrase that protects the private key. Note that this command will work even if the original key is not password protected. If the `-file` argument is not given, the default location of the file containing the private key is assumed:

- The location pointed to by `X509_USER_KEY`
- If `X509_USER_KEY` not set, `$HOME/.globus/userkey.pem`

Options

Table 53. Command line options

| | |
|----------------|---|
| help, -usage | Displays usage. |
| -version | Displays version. |
| -file location | Changes the passphrase on the key stored in the file at the non-standard location 'location'. |

Limitations

Nothing applicable

Name

grid-proxy-init -- Generate a new *proxy certificate*

grid-proxy-init

Tool description

grid-proxy-init generates X.509 proxy certificates.

By default, this command generates [RFC 3820](#)¹ Proxy Certificates.

There are also options available for generating other types of proxy certificates, including limited, independent and legacy. For more information about proxy certificate types and their compatibility in GT, see <http://dev.globus.org/wiki/Security/ProxyCertTypes>.

Command syntax

```
grid-proxy-init [-help][-pwstdin][-limited][-valid H:M] ...
```

¹ <http://www.ietf.org/rfc/rfc3820.txt>

Options

Table 54. Command line options

| | |
|-----------------------------------|---|
| -help, -usage | Displays usage. |
| -version | Displays version. |
| -debug | Enables extra debug output. |
| -q | Quiet mode, minimal output. |
| -verify | Verifies the certificate to make the proxy for. |
| -pwstdin | Allows passphrase from stdin. |
| -limited | Creates a limited globus proxy. |
| -independent | Creates an independent globus proxy. |
| -draft | Creates a draft (GSI-3) proxy. |
| -old | Creates a legacy globus proxy. |
| -valid <h:m> | Proxy is valid for <i>h</i> hours and <i>m</i> minutes (default:12:00). |
| -hours <hours> | Deprecated support of hours option. |
| -bits <bits> | Number of bits in key {512 1024 2048 4096}. |
| -policy <policyfile> | File containing the policy to store in the ProxyCertInfo extension. |
| -pl <oid>, -policy-language <oid> | OID string for the policy language used in the policy file. |
| -path-length <l> | Allows a chain of at most 1 proxies to be generated from this one. |
| -cert <certfile> | Non-standard location of user certificate. |
| -key <keyfile> | Non-standard location of user key. |
| -certdir <certdir> | Non-standard location of trusted cert directory. |
| -out <proxyfile> | Non-standard location of new proxy cert. |

Creating a Proxy Certificate

Proxies are certificates signed by the user, or by another proxy, that do not require a password to submit a job. They are intended for short-term use, when the user is submitting many jobs and cannot be troubled to repeat his password for every job.

The subject of a proxy certificate is the same as the subject of the certificate that signed it, with /CN=proxy added to the name. The gatekeeper will accept any job requests submitted by the user, as well as any proxies he has created.

Proxies provide a convenient alternative to constantly entering passwords, but are also less secure than the user's normal security credential. Therefore, they should always be user-readable only, and should be deleted after they are no longer needed (or after they expire).

To create a proxy with the default expiration (12 hours), run the grid-proxy-init program. For example:

```
% grid-proxy-init
```

The grid-proxy-init program can also take arguments to specify the expiration and proxy key length. For example:

```
% grid-proxy-init -hours 8 -bits 512
```

Limitations

Nothing applicable

Name

grid-proxy-destroy -- Destroy the current proxy certificate (previously created with grid-proxy-init)

grid-proxy-destroy

Tool description

grid-proxy-destroy removes X.509 proxy certificates.

Command syntax

```
grid-proxy-destroy [-help][--dryrun][-default][-all][--] [file1...]
```

Options

Table 55. Command line options

| | |
|-----------------|---|
| -help, -usage | Displays usage. |
| -version | Displays version. |
| -debug | Displays debugging information. |
| -dryrun | Prints what files would have been destroyed. |
| -default | Destroys file at default proxy location. |
| -all | Destroys any user (default) and delegated proxies that are found. |
| -- | Ends processing of options. |
| file1 file2 ... | Destroys the files listed. |

Limitations

Nothing applicable

Name

grid-proxy-info -- Display information obtained from a proxy certificate

grid-proxy-info

Tool description

grid-proxy-info extracts information from X.509 proxy certificates.

Command syntax

```
grid-proxy-info [-help][-f proxyfile][-subject][...][-e [-h H][-b B]]
```

Options

Table 56. Command line options

| | |
|------------------------|--|
| -help, -usage | Displays usage. |
| -version | Displays version. |
| -debug | Displays debugging output. |
| -file <proxyfile> (-f) | Non-standard location of proxy. |
| [printoptions] | See Table 57, “Print options”. |
| -exists [options] (-e) | Determine whether a valid proxy exists. <code>options</code> may contain any <u>validation options</u> described below. If a proxy exists, and meets any criteria defined by the validity options, then grid-proxy-info will terminate with the exit code 0. Otherwise, grid-proxy-info will terminate with the exit code 1. If no validity options are specified, the program will terminate with 0 if a currently-valid proxy file exists. |

Table 57. Print options

| | |
|---------------|---|
| -subject (-s) | Distinguished name (DN) of the subject. |
| -issuer (-i) | DN of the issuer (certificate signer). |
| -identity | DN of the identity represented by the proxy. |
| -type | Type of proxy (full or limited). |
| -timeleft | Time (in seconds) until proxy expires. |
| -strength | Key size (in bits). |
| -all | All above options in a human readable format. |
| -text | All of the certificate. |
| -path | Pathname of the proxy file. |

Table 58. Validity options

| | |
|-----------------|--|
| -valid H:M (-v) | Time requirement for the proxy to be valid. |
| -hours H (-h) | Time requirement for the proxy to be valid (deprecated, use -valid instead). |
| -bits B (-b) | Strength requirement for the proxy to be valid. |

Limitations

Nothing applicable

Name

grid-mapfile-add-entry -- Add an entry to a *grid map file*

grid-mapfile-add-entry

Tool description

grid-mapfile-add-entry adds entries to grid map files.

Command syntax

```
grid-mapfile-add-entry -dn DN -ln LN [-help] [-d] [-f mapfile FILE]
```

Options:

Table 59. Command line options

| | |
|------------------------|---|
| -help, -usage | Displays help. |
| -version | Displays version. |
| -dn DN | Distinguished Name (DN) to add. Remember to quote the DN if it contains spaces. |
| -ln LN1 [LN2...] | Local login name(s) to which the DN is mapped. |
| -dryrun, -d | Shows what would be done but will not add the entry. |
| -mapfile FILE, -f FILE | Path of the grid map file to be used. |

Limitations

Nothing applicable.

Name

grid-mapfile-check-consistency -- Check the internal consistency of a grid map file

grid-mapfile-check-consistency

Tool description

grid-mapfile-check-consistency checks that the given grid mapfile conforms to the expected format as well as checking for common subject name problems.

Command syntax

grid-mapfile-check-consistency [-help] [-mapfile FILE]

Options:

Table 60. Command line options

| | |
|------------------------|---------------------------------------|
| -help, -usage | Displays help. |
| -version | Displays version. |
| -mapfile FILE, -f FILE | Path of the grid map file to be used. |

Limitations

Nothing applicable

Name

grid-mapfile-delete-entry -- Delete an entry from a grid map file

grid-mapfile-delete-entry

Tool description

grid-mapfile-delete entry deletes a grid map file entry from the given file.

Command syntax

grid-mapfile-delete-entry [-help] [-dn <DN>] [-ln <local name>] [-d] [-f file]

Options:

Table 61. Command line options

| | |
|------------------------|---|
| -help, -usage | Displays help. |
| -version | Displays version. |
| -dn <DN> | Distinguished Name (DN) to delete. |
| -ln <local name> | Local Login Name (LN) to delete. |
| -dryrun, -d | Shows what would be done but will not delete the entry. |
| -mapfile file, -f file | Path of the grid map file to be used. |

Limitations

Nothing applicable.

CAS Query Commands

The CAS Query commands do not alter the state of the database and any CAS user who has cas/query permissions may use the commands to retrieve data from the CAS server.

The following queries can be run against the CAS server. These are typically used by CAS clients (who may not be administrators).

The user need cas/query permissions to perform these operations—that is, the user must have permission to query on the cas server object.

Name

cas-whoami -- Getting a user's CAS identity.

cas-whoami [*options*]

Tool description

The **cas-whoami** command returns the CAS user nick of the client.

Command options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• msg for GSI Secure Message, or• conv for GSI Secure Conversation. |
| -p, --protection <type> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• sig for signature, or• enc for encryption. |
| -s <i>cas-url</i> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . The instance URL typically looks like <code>http://Host:Port/wsrp/services/CASService</code> , where <i>Host</i> and <i>Port</i> are the host and port where the container with the CAS service is running. |
| -v | Prints the version number. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z <i>authorization</i> | Specifies the type of authorization used, such as <code>self</code> or <code>host</code> . |

If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value.

Alternatively, an environment variable can be set as shown [here](#).

If none of the above are set, host authorization is done by default and the expected server credential is `cas/<fqdn>`, where `<fqdn>` is the fully qualified domain name of the host on which the CAS service is up.



Note

If the service being contacted is using GSI Secure Transport , then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Name

cas-list-object -- Getting object list

cas-list-object [*options*] *type*

Tool description

The **cas-list-object** command returns a list of CasObjects in the database of the requested type.

Command Options

~~Use~~ Use one of the following to indicate the type of of CasObjects you want listed:

- trustAnchor
- user
- userGroup
- object
- objectGroup
- objectGroup
- namespace
- serviceType
- serviceAction
- serviceActionGp

Common Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |

- `-m, --securityMech <type>` Specifies the authentication mechanism. The value *type* can be:
- `msg` for GSI Secure Message, or
 - `conv` for GSI Secure Conversation.
- `-p, --protection <type>` Specifies the protection level. *type* can be:
- `sig` for signature, or
 - `enc` for encryption.
- `-s cas-url` Sets the CAS Service instance, where *cas-url* is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown [here](#).
- The instance URL typically looks like `http://Host:Port/wsrp/services/CASService`, where *Host* and *Port* are the host and port where the container with the CAS service is running.
- `-v` Prints the version number.
- `-x, --proxyFilename <value>` Sets the proxy file to use as client credential.
- `-z authorization` Specifies the type of authorization used, such as `self` or `host`.
- If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value.
- Alternatively, an environment variable can be set as shown [here](#).
- If none of the above are set, host authorization is done by default and the expected server credential is `cas/<fqdn>`, where `<fqdn>` is the fully qualified domain name of the host on which the CAS service is up.



Note

If the service being contacted is using GSI Secure Transport, then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Name

cas-get-object -- Getting CAS object

cas-get-object [*options*] *type name*

Tool description

The **cas-get-object** command returns the particular object of the said type and name.

Command Options

te Use one of the following to indicate the type of of CasObjects you want to get:

- trustAnchor
- user
- userGroup
- object
- objectGroup
- namespace
- serviceType
- serviceAction
- serviceActionGp

me Use one of the following to indicate the name of the specific CAS object you want to get:

- *nickname* (if getting trustAnchor, user, userGroup, or namespace)
- *objectNamespace objectName* (if getting object or objectGroup)
- *serviceTypeName* (if getting serviceType, serviceAction, or serviceActionGp)

Common Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |

| | |
|--|--|
| -f, --descriptor <i><file></i> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <i><value></i> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <i><type></i> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• <code>msg</code> for GSI Secure Message, or• <code>conv</code> for GSI Secure Conversation. |
| -p, --protection <i><type></i> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• <code>sig</code> for signature, or• <code>enc</code> for encryption. |
| -s <i>cas-url</i> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . The instance URL typically looks like <code>http://Host:Port/wsrp/services/CASService</code> , where <i>Host</i> and <i>Port</i> are the host and port where the container with the CAS service is running. |
| -v | Prints the version number. |
| -x, --proxyFilename <i><value></i> | Sets the proxy file to use as client credential. |
| -z <i>authorization</i> | Specifies the type of authorization used, such as <code>self</code> or <code>host</code> . If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value. Alternatively, an environment variable can be set as shown here . If none of the above are set, host authorization is done by default and the expected server credential is <code>cas/<fqdn></code> , where <i><fqdn></i> is the fully qualified domain name of the host on which the CAS service is up. |



Note

If the service being contacted is using GSI Secure Transport, then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Name

cas-group-list-entries -- Getting group members

cas-group-list-entries [*options*] *type name*

Tool description

The **cas-group-list-entries** command returns a list of group members.

Command Options

te Use one of the following to indicate the type of group for which you want a list of members:

- user
- object
- serviceType

me The name of the group.

Common Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• msg for GSI Secure Message, or• conv for GSI Secure Conversation. |
| -p, --protection <type> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• sig for signature, or• enc for encryption. |

- `-s cas-url` Sets the CAS Service instance, where `cas-url` is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown [here](#).
- The instance URL typically looks like `http://Host:Port/wsrf/services/CASService`, where `Host` and `Port` are the host and port where the container with the CAS service is running.
- `-v` Prints the version number.
- `-x, --proxyFilename <value>` Sets the proxy file to use as client credential.
- `-z authorization` Specifies the type of authorization used, such as `self` or `host`.
- If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value.
- Alternatively, an environment variable can be set as shown [here](#).
- If none of the above are set, host authorization is done by default and the expected server credential is `cas/<fqdn>`, where `<fqdn>` is the fully qualified domain name of the host on which the CAS service is up.



Note

If the service being contacted is using GSI Secure Transport , then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Name

cas-find-policies -- Getting policy information

```
cas-find-policies [options] [-c cas-url] type name
```

Tool description

The **cas-find-policies** command returns all applicable policies, both policies that are implicit to the CAS server and those that are external.

Command options

-c cas-url The URL of the CAS service.

type Use one of the following to indicate the type of CasObjects:

- trustAnchor
- user
- userGroup
- object
- objectGroup
- namespace
- serviceType
- serviceAction
- serviceActionGp

name Use the type of name corresponding to the appropriate CasObject:

- *nickname* (for trustAnchors, users, or namespaces)
- *groupName* (for userGroups, objectGroups, or serviceActionGps)
- *objectNamespace/objectName* (for objects)
- *serviceTypeName* (or) *serviceType/Action* (for serviceTypes or serviceActions)

Common Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

-a, --anonymous Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism.

| | |
|---|--|
| <code>-c, --serverCertificate <file></code> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| <code>-debug</code> | Runs the client with debug message traces and error stack traces. |
| <code>-f, --descriptor <file></code> | Specifies a client security descriptor. Overrides all other security settings. |
| <code>-help</code> | Prints the usage message for the client. |
| <code>-l, --contextLifetime <value></code> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| <code>-m, --securityMech <type></code> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• <code>msg</code> for GSI Secure Message, or• <code>conv</code> for GSI Secure Conversation. |
| <code>-p, --protection <type></code> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• <code>sig</code> for signature, or• <code>enc</code> for encryption. |
| <code>-s cas-url</code> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . The instance URL typically looks like <code>http://Host:Port/wsrp/services/CASService</code> , where <i>Host</i> and <i>Port</i> are the host and port where the container with the CAS service is running. |
| <code>-v</code> | Prints the version number. |
| <code>-x, --proxyFilename <value></code> | Sets the proxy file to use as client credential. |
| <code>-z authorization</code> | Specifies the type of authorization used, such as <code>self</code> or <code>host</code> . If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value. Alternatively, an environment variable can be set as shown here . If none of the above are set, host authorization is done by default and the expected server credential is <code>cas/<fqdn></code> , where <i><fqdn></i> is the fully qualified domain name of the host on which the CAS service is up. |



Note

If the service being contacted is using GSI Secure Transport, then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Name

query-cas-service -- Query CAS Service (using OGSA AuthZ interface)

query-cas-service [*options*] *assertionFilename*

Tool description

The **query-cas-service** command returns a SAML Response containing SAML Assertions with user rights for a given SAML Query. This client uses the OGSA AuthZ interface and writes out the retrieved assertion to a file.

Command options

~~⌘~~ File to write assertions to.

~~⌘~~

~~⌘~~

~~⌘~~

~~⌘~~

~~⌘~~

Common Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• msg for GSI Secure Message, or• conv for GSI Secure Conversation. |
| -p, --protection <type> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• sig for signature, or• enc for encryption. |

- `-s cas-url` Sets the CAS Service instance, where `cas-url` is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown [here](#).
The instance URL typically looks like `http://Host:Port/wsrp/services/CASService`, where `Host` and `Port` are the host and port where the container with the CAS service is running.
- `-v` Prints the version number.
- `-x, --proxyFilename <value>` Sets the proxy file to use as client credential.
- `-z authorization` Specifies the type of authorization used, such as `self` or `host`.
If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value.
Alternatively, an environment variable can be set as shown [here](#).
If none of the above are set, host authorization is done by default and the expected server credential is `cas/<fqdn>`, where `<fqdn>` is the fully qualified domain name of the host on which the CAS service is up.



Note

If the service being contacted is using GSI Secure Transport , then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

CAS Admin Commands

Name

cas-proxy-init -- Generate a CAS proxy

```
cas-proxy-init [common options] [ -p proxyfile | -t tag ]
```

Tool description

The **cas-proxy-init** command contacts a CAS server, obtains an assertion for the user, and embeds it in a credential. This credential can be used to access CAS-enabled services.

Options

Command-specific options

-b *policyFileName* Generate a CAS credential that includes only those permissions specified in file *policyFileName* (the default is to generate a credential with all the user's permissions). Details about the template of the file is provided [here](#).

-u *tag* Choose a filename in which to store the CAS credential based on the value *tag*. Cannot be used with the *-p* option.

-w *generatedCredentialFile* Specify the file in which to store the CAS credential. Cannot be used with the *-t* option.

Common Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |

| | |
|--|--|
| <code>-m, --securityMech <type></code> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• <code>msg</code> for GSI Secure Message, or• <code>conv</code> for GSI Secure Conversation. |
| <code>-p, --protection <type></code> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• <code>sig</code> for signature, or• <code>enc</code> for encryption. |
| <code>-s cas-url</code> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . The instance URL typically looks like <code>http://Host:Port/wsrp/services/CASService</code> , where <i>Host</i> and <i>Port</i> are the host and port where the container with the CAS service is running. |
| <code>-v</code> | Prints the version number. |
| <code>-x, --proxyFilename <value></code> | Sets the proxy file to use as client credential. |
| <code>-z authorization</code> | Specifies the type of authorization used, such as <code>self</code> or <code>host</code> . If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value. Alternatively, an environment variable can be set as shown here . If none of the above are set, host authorization is done by default and the expected server credential is <code>cas/<fqdn></code> , where <i><fqdn></i> is the fully qualified domain name of the host on which the CAS service is up. |



Note

If the service being contacted is using GSI Secure Transport, then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Usage

The following gets the assertion from the CAS server, generates a proxy with the assertion and writes it out to "casProxy".

```
cas-proxy-init -p casProxy
```

Requesting specific permissions from the CAS server

It is possible to request specific permissions from the CAS server using the `-f` option. This option causes **cas-proxy-init** to read a set of requested rights from a file.

This file should contain one or more resource identifiers:

Resource: *ResourceNamespace* | *ResourceName*

For each resource, there should be one or more action identifiers:

serviceType *action*

For example, if the client needed assertions for "file/read" service/action (permission) on two resources ("ftp://sample1.org" and "ftp://sample3.org", both in "FTPNamespace") but "directory/read" and "directory/write" permissions on the former resource only, the policy file should have the following entries:

Resource: FTPNamespace | ftp://sample1.org

file read

directory read

directory write

Resource: FTPNamespace | ftp://sample3.org

file read

To indicate any resource, the following wildcard notation should be used:

uri:samlResourceWildcard

To indicate any action, the following wildcard notation for *serviceType* and *action* should be used. Note that this should be the first (and clearly the only action) in the list of actions specified. All other actions in the list are ignored and if it is not the first, it is not treated as a wildcard.

uri:samlActionNSWildcard uri:samlActionWildcard

For example, if the client needs assertions for all resources and all actions, the policy file should look like:

Resource: uri:samlResourceWildcard

uri:samlActionNSWildcard uri:samlActionWildcard

If the client needs assertions for all actions on resource "FTPNamespace|ftp://sample1.org", the policy file should be as follows:

Resource: FTPNamespace | ftp://sample1.org

uri:samlActionNSWildcard uri:samlActionWildcard

Name

cas-wrap -- Runs program with CAS credentials

```
cas-wrap [common options] [ -p proxyfile | -t tag ]
```

Tool description

The **cas-wrap** command runs a grid-enabled program, causing it to use previously-generated CAS credentials.

This command invokes the given command with the given argument using the specified previously-generated CAS credential. For example:

```
casAdmin$ cas-wrap -t my-community gsincftp myhost.edu
```

will look for a credential generated by a previous execution of:

```
casAdmin$ cas-proxy-init -t my-community
```

and then set the environment to use that credential while running the command:

```
casAdmin$ gsincftp myhost.edu
```

The second form should be used if **cas-proxy-init** was run with the `-p` option. For example:

```
casAdmin$ cas-wrap -p /path/to/my/cas/credential gsincftp myhost.edu
```

will look for a credential generated by a previous execution of:

```
casAdmin$ cas-proxy-init -p /path/to/my/cas/credential
```

and then set the environment to use that credential while running the command:

```
casAdmin$ gsincftp myhost.edu
```

Options

Command-specific Options

`-p` Specify the file in which to store the CAS credential. Cannot be used with the `-t` option.

~~proxy~~
file

`-t` Choose a filename in which to store the CAS credential based on the value *tag*. Cannot be used with the `-p` option.

Common Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

`-a, --anonymous` Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism.

| | |
|---|--|
| <code>-c, --serverCertificate <file></code> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| <code>-debug</code> | Runs the client with debug message traces and error stack traces. |
| <code>-f, --descriptor <file></code> | Specifies a client security descriptor. Overrides all other security settings. |
| <code>-help</code> | Prints the usage message for the client. |
| <code>-l, --contextLifetime <value></code> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| <code>-m, --securityMech <type></code> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• <code>msg</code> for GSI Secure Message, or• <code>conv</code> for GSI Secure Conversation. |
| <code>-p, --protection <type></code> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• <code>sig</code> for signature, or• <code>enc</code> for encryption. |
| <code>-s cas-url</code> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . The instance URL typically looks like <code>http://Host:Port/wsrp/services/CASService</code> , where <i>Host</i> and <i>Port</i> are the host and port where the container with the CAS service is running. |
| <code>-v</code> | Prints the version number. |
| <code>-x, --proxyFilename <value></code> | Sets the proxy file to use as client credential. |
| <code>-z authorization</code> | Specifies the type of authorization used, such as <code>self</code> or <code>host</code> . If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value. Alternatively, an environment variable can be set as shown here . If none of the above are set, host authorization is done by default and the expected server credential is <code>cas/<fqdn></code> , where <code><fqdn></code> is the fully qualified domain name of the host on which the CAS service is up. |



Note

If the service being contacted is using GSI Secure Transport, then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Usage

Example of using cas-wrap to transfer a file.

```
cas-wrap -p casProxy globus-url-copy gsiftp://somehost.edu/some_file_path \  
file:///some_file_path
```

Name

cas-enroll -- Enroll a CAS Object

```
cas-enroll [common options] trustAnchor userGpName nickname authMethod authData cas-enroll [common options] namespace userGpName nickname basename comparisonAlg cas-enroll [common options] object userGpName objectName namespaceNick cas-enroll [common options] serviceType userGpName serviceTypeName
```

Tool description

This command line client is used to enroll a CAS Object, which includes trust anchors, namespaces, objects and service types.

Enrolling Trust Anchors

To enroll a trust anchor, the user must have `cas/enroll_trustAnchor` permission on that CAS server object (that is, the user must have permission to perform the `enroll_trustAnchor` action on the CAS service type).

The enroll operation allows the user to choose a user group to which `cas/grantAll` permission on the enrolled object should be granted. The nickname should be unique across the CAS database and is used to refer to this trust anchor.

To enroll trust anchors:

```
casAdmin$ cas-enroll [common options] trustAnchor userGpName nickname authMethod authData
```

where:

userGpName Indicates the user group to which `cas/grantAll` permission should be granted on this trust anchor entity.

nickname Indicates the trust anchor nickname.

authMethod Indicates the authentication method used by the trust anchor.

authData Indicates the data used for authentication, typically the DN.

Enrolling Namespaces

To enroll a namespace, the user must have `cas/enroll_namespace` permission (that is, the user must have permission to perform the `enroll_namespace` action on the cas service type).

The enroll operation allows the user to choose a userGroup to have `cas/grantAll` permission on the enrolled object. The comparison algorithm specified should be the name of the Comparison class that needs to be used to compare objects that belong to this namespace. The nickname should be unique across the CAS database and is used to refer to this user.

Also, two namespaces are added to the CAS database at boot up time, other than the inherent CAS Namespace:

- `FTPDirectoryTree` uses the `WildcardComparison` Algorithm and has the base URL set to the current directory.
- `FTPEXact` uses the `ExactComparison` Algorithm and has the base URL set to the current directory.

To enroll namespaces:

```
casAdmin$ cas-enroll [common options] namespace userGpName nickname basename comparisonAlg
```

where:

| | |
|----------------------|---|
| <i>userGpName</i> | Indicates the user group to which cas/grantAll permission should be granted on this trust anchor entity. |
| <i>nickname</i> | Indicates the nickname of the namespace to be unenrolled. If the trust anchor nickname specified does not exist, an error is <i>not</i> thrown. If the unenroll operation is successful, all policy data on that trust anchor is purged. |
| <i>basename</i> | Indicates the base URL for the namespace. |
| <i>comparisonAlg</i> | Indicates the comparison algorithm to be used. Unless the standard comparison algorithms described below are used, the fully qualified name of the class that needs to be used should be given. The class needs to extend from the abstract class <code>org.globus.cas.impl.service.ObjectComparison</code> . |

The two comparison classes provided as a part of the distribution are:

- `ExactComparison`: This class does a case-sensitive exact comparison of the object names. If `comparisonAlg` in the above method is set to `ExactComparison`, the class in the distribution is loaded and used.
- `WildcardComparison`: This class does wild card matching as described in [CAS Simple Policy Language](#)¹. It assumes that the wild card character is "*" and that the file separator is "/". If `comparisonAlg` in the above method is set to `WildcardComparison`, the class in the distribution is loaded and used.

Enrolling Objects

To enroll an object, the user must have cas/enroll_object permission (that is, the user must have permission to perform the enroll_object action on the cas service type).

The enroll operation allows the user to choose a userGroup to have cas/grantAll permission on the enrolled object. The name of the object and the namespace this object belongs to identify an object in the database and should be unique across the CAS database.

To enroll objects:

```
casAdmin$ cas-enroll [common options] object userGpName objectName namespaceNick
```

where:

| | |
|----------------------|--|
| <i>userGpName</i> | Indicates the user group to which cas/grantAll permission should be granted on this trust anchor entity. |
| <i>objectName</i> | Indicates the name of the object. |
| <i>namespaceNick</i> | Indicates the nickname of the namespace to which this object belongs. |

¹ http://www.globus.org/toolkit/docs/3.2/cas/CAS_policy_language_0.2.pdf

Enrolling Service Types

To enroll a service type, the user must have `cas/enroll_serviceType` permission (that is, the user must have permission to perform the `enroll_serviceType` action on the cas service type).

The enroll operation allows the user to choose a `userGroup` to have `cas/grantAll` permission on the enrolled service type. The service type name should be unique across the CAS database.

To enroll service types:

```
casAdmin$ cas-enroll [common options] serviceType userGpName serviceTypeName
```

where:

`userGpName` Indicates the user group to which `cas/grantAll` permission should be granted on this trust anchor entity.

`serviceTypeName` Indicates the service type name.

Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|---|---|
| <code>-a, --anonymous</code> | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| <code>-c, --serverCertificate <file></code> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| <code>-debug</code> | Runs the client with debug message traces and error stack traces. |
| <code>-f, --descriptor <file></code> | Specifies a client security descriptor. Overrides all other security settings. |
| <code>-help</code> | Prints the usage message for the client. |
| <code>-l, --contextLifetime <value></code> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| <code>-m, --securityMech <type></code> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none"> • <code>msg</code> for GSI Secure Message, or • <code>conv</code> for GSI Secure Conversation. |
| <code>-p, --protection <type></code> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none"> • <code>sig</code> for signature, or • <code>enc</code> for encryption. |
| <code>-s cas-url</code> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . |

The instance URL typically looks like `http://Host:Port/wsrp/services/CASService`, where *Host* and *Port* are the host and port where the container with the CAS service is running.

- `-v` Prints the version number.
- `-x, --proxyFilename <value>` Sets the proxy file to use as client credential.
- `-z authorization` Specifies the type of authorization used, such as `self` or `host`.
- If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value.
- Alternatively, an environment variable can be set as shown [here](#).
- If none of the above are set, host authorization is done by default and the expected server credential is `cas/<fqdn>`, where `<fqdn>` is the fully qualified domain name of the host on which the CAS service is up.



Note

If the service being contacted is using GSI Secure Transport , then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Usage

For detailed examples of using this command, see [Chapter 6, Example of CAS Server Administration](#) .

Name

cas-remove -- Remove a CAS object from the database

```
cas-remove [common options] trustAnchor nickname cas-remove [common options] namespace
nickname cas-remove [common options] object objName namespaceNick cas-remove [common
options] serviceType serviceTypeName
```

Tool description

Removing Trust Anchors

To remove a trust anchor, the user must have cas/remove permission on that trust anchor. The trust anchor must also be unused (that is, there may not be any users in the database that have this trust anchor or it may not be a part of any object group).

To remove trust anchors:

```
casAdmin$ cas-remove [options] trustAnchor nickname
```

where:

nickname Indicates the nickname of the trust anchor to be unenrolled.

If the trust anchor nickname specified does not exist, an error is *not* thrown. If the unenroll operation is successful, all policy data on that trust anchor is purged.

Removing Namespaces

To remove a namespace, the user must have cas/remove permission on that namespace. The namespace must also be unused — that is, there may not be any object in the database that belongs to this namespace.

```
casAdmin$ cas-remove [options] namespace nickname
```

where:

nickname Indicates the nickname of the namespace to be unenrolled.

If the namespace nickname specified does not exist, an error is *not* thrown. If the remove operation is successful, all policy data on that trust anchor is purged.

Removing Objects

To remove an object the user must have cas/remove permission on that object. The object must also be unused — that is, there may not be any object group in the database that this object belongs to.

```
casAdmin$ cas-remove [options] object objName namespaceNick
```

where:

objName Indicates the name of the object to be removed.

namespaceNick Indicates the nickname of the namespace to which this object belongs.

If the object specified does not exist, an error is *not* thrown. If the remove operation is successful, all policy data on that object is purged.

Removing Service Types

To remove a service type the user must have cas/remove permission on that service type. The service type must also be unused — that is, there may not be any service type to action mapping.

```
casAdmin$ cas-remove [options] serviceType serviceTypeName
```

where:

serviceTypeName Indicates the service type name.

If the service type specified does not exist, an error is *not* thrown. If the remove operation is successful, all policy data on that service type is purged.

Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• msg for GSI Secure Message, or• conv for GSI Secure Conversation. |
| -p, --protection <type> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• sig for signature, or• enc for encryption. |
| -s <i>cas-url</i> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . The instance URL typically looks like <code>http://Host:Port/wsrf/services/CASService</code> , where <i>Host</i> and <i>Port</i> are the host and port where the container with the CAS service is running. |

- `-v` Prints the version number.
- `-x, --proxyFilename <value>` Sets the proxy file to use as client credential.
- `-z authorization` Specifies the type of authorization used, such as `self` or `host`.
- If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value.
- Alternatively, an environment variable can be set as shown [here](#).
- If none of the above are set, host authorization is done by default and the expected server credential is `cas/<fqdn>`, where `<fqdn>` is the fully qualified domain name of the host on which the CAS service is up.



Note

If the service being contacted is using GSI Secure Transport , then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Name

cas-action -- Maintains service types

```
cas-action [common_options] [ add | remove ] serviceTypeName actionName
```

Tool description

Use the **cas-action** command to add an action mapping to a service type or remove an action mapping from a service type.

To add an action mapping to a service type, the user must have `cas/create_group_entry` permission on the service type.

To remove a service type action mapping, the user must have `cas/delete_group_entry` permission on the service type.

If the group member being removed does not exist, an error is *not* thrown.

Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• <code>msg</code> for GSI Secure Message, or• <code>conv</code> for GSI Secure Conversation. |
| -p, --protection <type> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• <code>sig</code> for signature, or• <code>enc</code> for encryption. |
| -s <i>cas-url</i> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . |

The instance URL typically looks like `http://Host:Port/wsrp/services/CASService`, where *Host* and *Port* are the host and port where the container with the CAS service is running.

- `-v` Prints the version number.
- `-x, --proxyFilename <value>` Sets the proxy file to use as client credential.
- `-z authorization` Specifies the type of authorization used, such as `self` or `host`.
- If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value.
- Alternatively, an environment variable can be set as shown [here](#).
- If none of the above are set, host authorization is done by default and the expected server credential is `cas/<fqdn>`, where `<fqdn>` is the fully qualified domain name of the host on which the CAS service is up.



Note

If the service being contacted is using GSI Secure Transport, then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Usage

For an example of using this command, see [Section 9, “Adding action mappings”](#).

Name

cas-group-admin -- Maintains user groups, object groups, or serviceAction groups

```
cas-group-admin [common options] [ user | object | serviceAction ] create userGpName groupName cas-  
group-admin [common options] [ user | object | serviceAction ] delete groupName
```

Tool description

Use **cas-group-admin** to create or delete user groups, object groups, or serviceAction groups. Note: to add or delete entries to these groups, see [\[fixme olink to other clients\]](#).

Adding user groups

To create a new user group the user must have cas/create_user_group permission (that is, the user must have permission to perform the create_user_group action on the cas service type). The user group name should be unique across the CAS database. The create operation allows the user to choose a user group to have cas/grantAll permission on the created user group. If the user group that is chosen to have cas/grantAll permission is the new group created, then the user making this request is added to the new group.

To add a user group:

```
casAdmin$ cas-group-admin [common options] user create userGpName groupName
```

where:

userGpName Indicates the user group to which cas/grantAll permission should be granted on this trust anchor entity.

groupName Indicates the name of the user group being created.

Deleting user groups

To delete a user group, the user must have cas/delete_user_group entry permission on that user group. The group must be empty and also must not be referenced from other entities in the database (for example, it should not be a member of some object group).

If the user group specified does not exist, an error is *not* thrown. If the delete operation is successful, all policy data on that user group is purged.

```
casAdmin$ cas-group-admin [common options] user delete groupName
```

where:

groupName Indicates the name of the user group to be deleted.

Creating An Object Group

To create a new object group, the user must have cas/create_object_group permission (that is, the user must have permission to perform the create_object_group action on the CAS service type). The object group name should be unique across the CAS database. The create operation allows the user to choose a user group to have cas/grantAll permission on the created object group.

```
casAdmin$ cas-group-admin [common options] object create userGpName groupName
```

where:

userGpName Indicates the user group to which cas/grantAll permission should be granted on this object group.

groupName Indicates the object group name.

Deleting An Object Group

To delete an object group, the user must have cas/delete_user_group entry permission on that object group. The group must be empty.

If the object group specified does not exist, an error is *not* thrown. If the delete operation is successful, all policy data on that object group is purged.

```
casAdmin$ cas-group-admin [common options] object delete groupName
```

where:

groupName The name of the object group to be deleted.

Creating A Service/Action Group

To create a new service/action group, the user must have cas/create_serviceAction_group permission (that is, the user must have permission to perform the create_serviceAction_group action on the CAS service type). The serviceAction group name should be unique across the CAS database. The create operation allows the user to choose a user group to have cas/grantAll permission on the created serviceAction group.

```
casAdmin$ cas-group-admin [common options] serviceAction create userGpName groupName
```

where:

userGp-Name Indicates the user group to which cas/grantAll permission should be granted on this service/action group.

groupName Indicates the name of the service/action group being created.

Deleting A Service/Action Group

To delete a service/action group, the user must have cas/delete_user_group entry permission on that service/action group. The group must be empty and also must not be referenced from any other entity in the database. For example, it should not be a member of some object group.

If the service/action group specified does not exist, an error is *not* thrown. If the delete operation is successful, all policy data on that service/action group is purged.

```
casAdmin$ cas-group-admin [common options] serviceAction delete groupName
```

where:

~~gp~~ Indicates the name of the service/action group to be deleted.

~~name~~

Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• msg for GSI Secure Message, or• conv for GSI Secure Conversation. |
| -p, --protection <type> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• sig for signature, or• enc for encryption. |
| -s cas-url | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . The instance URL typically looks like <code>http://Host:Port/wsrf/services/CASService</code> , where <i>Host</i> and <i>Port</i> are the host and port where the container with the CAS service is running. |
| -v | Prints the version number. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -z authorization | Specifies the type of authorization used, such as <code>self</code> or <code>host</code> . If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value. Alternatively, an environment variable can be set as shown here . If none of the above are set, host authorization is done by default and the expected server credential is <code>cas/<fqdn></code> , where <i><fqdn></i> is the fully qualified domain name of the host on which the CAS service is up. |



Note

If the service being contacted is using GSI Secure Transport , then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Usage

For examples of using this command, see [Chapter 6, Example of CAS Server Administration](#).

Name

cas-group-add-entry -- Adds CAS objects to CAS groups

```
cas-group-add-entry [common options] user groupName nickname cas-group-add-entry  
[common options] object groupName objectSpecDesc objectSpec cas-group-add-entry [common  
options] serviceAction groupName serviceTypeName actionName
```

Tool description

Use **cas-group-add-entry** to add users to a user group, objects to an object group, or service/actions to service/action groups. Note: to add or delete groups, see [fixme olink to other clients].

Adding Member To A User Group

To add a user to a user group, the user must have cas/add_group_entry permission on that particular user group. Only user nicknames that exist in the CAS database can be valid members.

```
casAdmin$ cas-group-add-entry [common options] user groupName nickname
```

where:

~~gp~~ Indicates the user group name to which the member needs to be added.

~~nk~~

~~nk~~ Indicates the nickname of the user to be added to this group.

~~nk~~

Adding Member To An Object Group

To add a member (an object group can have the following CasObjects as members: object, user, user group, service type, namespace or trust anchor) to an object group, the user must have cas/add_group_entry permission on that particular object group.

```
casAdmin$ cas-group-add-entry [common options] object groupName objectSpecDesc objectSpec
```

where:

~~gp~~ Indicates the object group name to which the member needs to be added.

~~nk~~

~~ob~~ Indicates the type of CasObject. Can be one of the following options:

~~ob~~

~~ob~~ • trustAnchor

~~ob~~

• user

• userGroup

• object

• namespace

• serviceType

~~o~~ Indicates the identifier for the CasObject the user is adding. Can be one of the following:

- ~~o~~ • nickname if adding a trustAnchor or user
- groupName if adding a userGroup
- objectNamespace objectName if adding an object
- nickname if adding a namespace
- serviceTypeName if adding a serviceType

Adding Service/Action To A Service/Action Group

To add a service/action to a serviceAction group, the user must have cas/add_group_entry permission on that particular serviceAction group (that is, the user must have permission to perform add_group_entry action on that service action group).

```
casAdmin$ cas-group-add-entry [common options] serviceAction groupName serviceTypeName act
```

where:

~~o~~ Indicates the service/action group to which the service/action needs to be added.

~~o~~

~~o~~ Indicates the service type name part of the mapping to be added to the group.

~~o~~

~~o~~

~~o~~

~~o~~ Indicates the action name part of the mapping to be added to the group.

~~o~~

~~o~~

Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. The value <i>type</i> can be: |

| | |
|--|--|
| | <ul style="list-style-type: none">• <code>msg</code> for GSI Secure Message, or• <code>conv</code> for GSI Secure Conversation. |
| <code>-p, --protection <type></code> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• <code>sig</code> for signature, or• <code>enc</code> for encryption. |
| <code>-s cas-url</code> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . The instance URL typically looks like <code>http://Host:Port/wsrp/services/CASService</code> , where <i>Host</i> and <i>Port</i> are the host and port where the container with the CAS service is running. |
| <code>-v</code> | Prints the version number. |
| <code>-x, --proxyFilename <value></code> | Sets the proxy file to use as client credential. |
| <code>-z authorization</code> | Specifies the type of authorization used, such as <code>self</code> or <code>host</code> . If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value. Alternatively, an environment variable can be set as shown here . If none of the above are set, host authorization is done by default and the expected server credential is <code>cas/<fqdn></code> , where <i><fqdn></i> is the fully qualified domain name of the host on which the CAS service is up. |



Note

If the service being contacted is using GSI Secure Transport, then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Usage

For examples of using this command, see [Chapter 6, Example of CAS Server Administration](#).

Name

cas-group-remove-entry -- Removing CAS objects from CAS groups

```
cas-group-remove-entry [common options] user groupName nickname cas-group-remove-  
entry [common options] object groupName objectSpec objectSpecDesc cas-group-remove-  
entry [common options] serviceAction groupName serviceTypeName actionName
```

Tool description

Use **cas-group-remove-entry** to remove users from a user group, objects from an object group, or service/actions from a service/action group. Note: to add or delete groups, see [fixme olink to other clients].

Removing User From A User Group

To remove a user from a user group, the user must have `cas/remove_group_entry` permission on that particular user group.

If the group member being removed does not exist, an error is *not* thrown.

```
casAdmin$ cas-group-remove-entry [common options] user groupName nickname
```

where:

~~gp~~ Indicates the user group name from which the member needs to be removed.

~~nk~~

~~nk~~ Indicates the nickname of the user to be removed from this group.

~~nk~~

Removing Member From An Object Group

To remove an object from an object group the user must have `cas/remove_group_entry` permission on that particular object group:

If the group member being removed does not exist, an error is *not* thrown.

```
casAdmin$ cas-group-remove-entry [common options] object groupName objectSpec objectSpecDesc
```

where:

~~gp~~ Indicates the object group name from which the member needs to be removed.

~~nk~~

~~ob~~ Indicates the type of CasObject. Can be one of the following options:

~~ob~~

~~ob~~ • trustAnchor

~~ob~~

• user

• userGroup

• object

• namespace

- `serviceType`

~~o~~ Indicates the identifier for the CasObject the user is adding. Can be one of the following:

- ~~o~~ • `nickname` if adding a trustAnchor or user
- `groupName` if adding a userGroup
- `objectNamespace objectName` if adding an object
- `nickname` if adding a namespace
- `serviceTypeName` if adding a serviceType

Removing A Service/Action From A Service/Action Group

To remove a service/action from a service/action group, the user must have `cas/remove_group_entry` permission on that particular service/action group.

If the action being removed does not exist, an error is *not* thrown.

```
casAdmin$ cas-group-remove-entry [common options] serviceAction groupName serviceTypeName
```

where:

~~o~~ Indicates the serviceAction group name from which the service/action needs to be removed.
~~o~~

~~o~~ Indicates the service type name part of the mapping to be removed from the group.
~~o~~
~~o~~
~~o~~

~~o~~ Indicates the action name part of the mapping to be removed from the group.
~~o~~
~~o~~

Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|---|--|
| <code>-a, --anonymous</code> | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| <code>-c, --serverCertificate <file></code> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| <code>-debug</code> | Runs the client with debug message traces and error stack traces. |
| <code>-f, --descriptor <file></code> | Specifies a client security descriptor. Overrides all other security settings. |
| <code>-help</code> | Prints the usage message for the client. |

| | |
|--|--|
| <code>-l, --contextLifetime <value></code> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| <code>-m, --securityMech <type></code> | Specifies the authentication mechanism. The value <i>type</i> can be: <ul style="list-style-type: none">• <code>msg</code> for GSI Secure Message, or• <code>conv</code> for GSI Secure Conversation. |
| <code>-p, --protection <type></code> | Specifies the protection level. <i>type</i> can be: <ul style="list-style-type: none">• <code>sig</code> for signature, or• <code>enc</code> for encryption. |
| <code>-s cas-url</code> | Sets the CAS Service instance, where <i>cas-url</i> is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown here . The instance URL typically looks like <code>http://Host:Port/wsrp/services/CASService</code> , where <i>Host</i> and <i>Port</i> are the host and port where the container with the CAS service is running. |
| <code>-v</code> | Prints the version number. |
| <code>-x, --proxyFilename <value></code> | Sets the proxy file to use as client credential. |
| <code>-z authorization</code> | Specifies the type of authorization used, such as <code>self</code> or <code>host</code> . If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value. Alternatively, an environment variable can be set as shown here . If none of the above are set, host authorization is done by default and the expected server credential is <code>cas/<fqdn></code> , where <code><fqdn></code> is the fully qualified domain name of the host on which the CAS service is up. |



Note

If the service being contacted is using GSI Secure Transport, then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Name

cas-rights-admin -- Granting or revoking permissions

```
cas-rights-admin [common options] [ grant | revoke ] userGroupName objectSpecDesc objectSpec  
actionSpecDesc actionSpec
```

Tool description

Use **cas-rights-admin** to grant or revoke rights.

Granting Permissions To A User Group On An Object/Object Group

The user may grant permissions to a user group on an object or object group to perform a service action or service action group (that is, to perform any action that is a member of the service action group to which permission is granted), provided the user has both:

- cas/grant permission on the object or object group, and
- permission to perform the service action or service action group on the object or object group.

```
casAdmin$ cas-rights-admin [common options] grant userGroupName objectSpecDesc objectSpec
```

where:

~~⌘~~ Indicates the user group to be granted permission.

~~⌘~~
~~⌘~~

~~⌘~~ Indicates the identifier for the object or object group.

~~⌘~~
~~⌘~~

~~⌘~~ Indicates the type:

- ~~⌘~~
- object
 - objectGroup

~~⌘~~ Indicates the identifier for action or action group.

~~⌘~~
~~⌘~~

~~⌘~~ Indicates the type:

- ~~⌘~~
- serviceAction
 - serviceActionGp

Revoking A Policy In The CAS Database

The user may revoke a policy in the CAS database if the user has cas/revoke permission on the object or object group on which the policy is defined.

```
casAdmin$ cas-rights-admin [common options] revoke userGroupName objectSpecDesc objectSpec
```

where:

~~⌘~~ Indicates the user group for which you want to grant permission.

~~⌘~~
~~⌘~~

~~⌘~~ Indicates the type of CasObject. Can be one of the following:

~~⌘~~
~~⌘~~
~~⌘~~

- trustAnchor
- user
- userGroup
- object
- namespace
- serviceType
- userGroup

~~⌘~~ Indicates the identifier for the object or object group.

~~⌘~~
~~⌘~~

~~⌘~~ Indicates the identifier for the action or action group.

~~⌘~~
~~⌘~~

~~⌘~~ Indicates the type (serviceAction or serviceActionGp).

~~⌘~~
~~⌘~~
~~⌘~~

Options

Important

If you have an asterisk (*) in your command, you might need to escape it with a backslash (\).

| | |
|--------------------------------|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. <i>value</i> is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |

- `-m, --securityMech <type>` Specifies the authentication mechanism. The value *type* can be:
- `msg` for GSI Secure Message, or
 - `conv` for GSI Secure Conversation.
- `-p, --protection <type>` Specifies the protection level. *type* can be:
- `sig` for signature, or
 - `enc` for encryption.
- `-s cas-url` Sets the CAS Service instance, where *cas-url* is the URL of the CAS service instance. Alternatively, an environment variable can be set as shown [here](#).
- The instance URL typically looks like `http://Host:Port/wsrp/services/CASService`, where *Host* and *Port* are the host and port where the container with the CAS service is running.
- `-v` Prints the version number.
- `-x, --proxyFilename <value>` Sets the proxy file to use as client credential.
- `-z authorization` Specifies the type of authorization used, such as `self` or `host`.
- If you cannot use a standard method for authorization, you can use the specific CAS server's identity as the value.
- Alternatively, an environment variable can be set as shown [here](#).
- If none of the above are set, host authorization is done by default and the expected server credential is `cas/<fqdn>`, where *<fqdn>* is the fully qualified domain name of the host on which the CAS service is up.



Note

If the service being contacted is using GSI Secure Transport, then the container credentials configured for the service will be used, even if service/resource level credentials are configured. Hence authorization needs to be done based on the DN of the container credentials.

Usage

For an example of using this command, see [Chapter 6, Example of CAS Server Administration](#).

Delegation Service Commands

Name

globus-credential-delegate -- Delegation client

globus-credential-delegate

Tool description

Used to contact a Delegation Factory Service and store a delegated credential. A delegated credential is created and stored in a delegated credential WS-Resource, and the Endpoint Reference(EPR) of the credential is written out to a file for further use.

Command syntax

globus-credential-delegate [options] <eprFilename>

Table 62. globus-credential-delegate options

| | |
|---|---|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -g, --delegation <mode> | Enables delegation. mode can be either ' limited ' or ' full '. Only supported with the GSI Secure Conversation authentication mechanism. |
| -help | Prints the usage message for the client. |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -x, --proxyFilename <value> | Sets the proxy file to use as the client credential. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be ' msg ' for GSI Secure Message, or ' conv ' for GSI Secure Conversation. |
| -p, --protection <type> | Specifies the protection level. type can be ' sig ' for signature or ' enc ' for encryption. |
| -s, --service <url> | Specifies the Delegation Factory Service URL. |
| -x, --proxyFilename <value> | Sets the proxy file to use as client credential. |
| -y, --lifetine <value> | Lifetime of delegated credential in seconds. Default is 43200 (which is 12 hours). |
| -z, --authorization <type> | Specifies authorization type. type can be ' self ', ' host ', ' none ', or a string specifying the expected identity of the remote party. |
| <eprFilename> | Filename to write the EPR of delegated credential to. |

Name

`globus-credential-refresh -- Delegation refresh client`

`globus-credential-refresh`

Tool description

Used to refresh delegated credentials pointed to by the specified EPR. A new credential is generated and the one previously created by the Delegation Service is overwritten.

Command syntax

`globus-credential-refresh [options]`

Table 63. globus-credential-refresh options

| | |
|---|--|
| -a, --anonymous | Enables anonymous authentication. Only supported with transport security or the GSI Secure Conversation authentication mechanism. |
| -c, --serverCertificate <file> | Specifies the server's <i>certificate</i> file used for encryption. Only needed for the GSI Secure Message authentication mechanism. |
| -debug | Runs the client with debug message traces and error stack traces |
| -e, --eprFile <file> | Specifies an <i>XML</i> file that contains the <i>WS-Addressing</i> endpoint reference. The EPR would be of the delegation resource that is to be refreshed. |
| -f, --descriptor <file> | Specifies a client security descriptor. Overrides all other security settings. |
| -g, --delegation <mode> | Enables delegation. mode can be either ' limited ' or ' full '. Only supported with the GSI Secure Conversation authentication mechanism. |
| -help | Prints the usage message for the client. |
| -k, --key <name value> | Specifies the resource key. The name is the QName of the resource key in the string form: {namespaceURI}localPart , where localPart is the simple value of the key. For complex keys, use the --eprFile option. For Delegation resource, the name will be as specified in the <i>delegationResourceKey</i> element and will replace <i>delegationResourceKey</i> with the actual key: -k " {http://www.globus.org/08/2004/delegationService}DelegationKey delegationResourceKey |
| -l, --contextLifetime <value> | Sets the lifetime of the client security context. value is in milliseconds. Only supported with the GSI Secure Conversation authentication mechanism. |
| -m, --securityMech <type> | Specifies the authentication mechanism. type can be ' msg ' for GSI Secure Message, or ' conv ' for GSI Secure Conversation. |
| -p, --protection <type> | Specifies the protection level. type can be ' sig ' for signature or ' enc ' for encryption. |
| -s, --service <url> | Specifies the Delegation Factory Service URL. |
| -x, --proxyFileName <value> | Sets the proxy file to use as the client credential. |
| -y, --lifetime <value> | Lifetime of delegated credential in seconds. Defaults to 43200 (which is 12 hours). |
| -z, --authorization <type> | Specifies authorization type. type can be ' self ', ' host ', ' none ', or a string specifying the expected identity of the remote party. |

Name

globus-delegation-client -- C Delegation client

```
globus-delegation-client [OPTION...] {SERVICE-SPECIFIER} {{EPR-FILENAME} | {-refresh}}
```

Description

Create or refresh delegated credentials in a service container. If the `-refresh` option is specified on the command-line, then the credential associated with an existing `DelegationService` resource is updated with a new credential. Otherwise, the `SERVICE-SPECIFIER` is interpreted as a `DelegationFactoryService` and a new `DelegationService` resource is created.

Command syntax

```
globus-delegation-client [OPTION...] {SERVICE-SPECIFIER} {{EPR-FILENAME} | {-refresh}}
```

`SERVICE-SPECIFIER`: [-s URI [-k KEY VALUE] | -e FILENAME]

`EPR-FILENAME`: Name of file to store EPR of new delegated credential.

Table 64. Common options

| | |
|--|--|
| -a --anonymous | Use anonymous authentication. Requires either -m 'conv' or transport (https) security. |
| -d, --debug | Enables debug mode. In debug mode, all SOAP messages will be displayed to stderr and full WSRF Fault messages will be displayed. |
| -e --eprFile FILENAME | Load service EPR from FILENAME. This EPR is used to contact the WSRF service. |
| -h --help | Displays help information about the command. |
| -k --key KEYNAME VALUE | Set resource key in the service EPR to be named KEYNAME with VALUE as its value. This can be combined with -s to construct an EPR without having an xml file on hand. The KEYNAME is a QName string in the format {namespaceURI}localPart . while the VALUE is a literal string to place in the element. For example, the option -k '{http://www.globus.org}MyKey' 128 would be rendered as <MyKey xmlns="http://www.globus.org">128</MyKey> |
| -m, --securityMech TYPE | Set authentication mechanism. TYPE is one of msg for WS-SecureMessage or conv for WS-SecureConversation. |
| -p, --protection LEVEL | Set message protection level. LEVEL is one of sig for digital signature or enc for encryption. The default is 'sig'. |
| -s --service ENDPOINT | Set ENDPOINT the service URL to use. Will be composed with the -k parameter if present to add ReferenceProperties to the ENDPOINT |
| -t --timeout SECONDS | Set client timeout to SECONDS. |
| -u --usage | Print short usage message. |
| -V --version | Show version information and exit. |
| -v --certKeyFiles CERTIFICATE-FILENAME KEY-FILENAME | Use credentials located in CERTIFICATE-FILENAME and KEY-FILENAME . The key file must be unencrypted. |
| -x --proxyFilename FILENAME | Use proxy credentials located in FILENAME . |
| -z --authorization TYPE | Set authorization mode. TYPE can be self , host , none , or a string specifying the identity of the remote party. The default is self . |
| --versions | Show version information for all loaded modules and exit. |

Table 65. Application-specific options

| | |
|-------------------------------|--|
| -g --delegation MODE | Set the delegation mode. MODE can be 'limited' or 'full'. The default is 'limited' |
| -r --refresh | Refresh a credential instead of creating a new delegated credential resource. |

Examples

Create a new delegated credential resource and store the EPR of the resource in `~/ .globus/delegation.epr`:

```
% globus-delegation-client -z host -s https://gridhost.virtual.org:8443/wsrf/services/Dele
```

Refresh the previously delegated credential:

```
% globus-delegation-client -z host -e ~/delegation.epr -refresh
```

Destroy the delegated credential:

```
% globus-wsrf-destroy -z host -e ~/delegation.epr
```

GridFTP Commands

Name

globus-url-copy -- Multi-protocol data movement

globus-url-copy

Tool description

globus-url-copy is a scriptable command line tool that can do multi-protocol data movement. It supports gsiftp:// (GridFTP), ftp://, http://, https://, and file:/// protocol specifiers in the URL. For GridFTP, globus-url-copy supports all implemented functionality. Versions from GT 3.2 and later support file globbing and directory moves.

- [Before you begin](#)
- [Command syntax](#)
- [Command line options](#)
 - [Informational options](#)
 - [Utility options](#)
 - [Reliability options](#)
 - [Performance options](#)
 - [Security-related options](#)
- [Default usage](#)
- [MODES in GridFTP](#)
- [If you run a GridFTP server by hand](#)
- [How do I choose a value for the TCP buffer size \(-tcp-bs\) option?](#)
- [How do I choose a value for the parallelism \(-p\) option?](#)
- [Limitations](#)
- [Interactive clients for GridFTP](#)

Before you begin

Important

To use gsiftp:// and https:// protocols, you must have a [certificate](#) to use globus-url-copy. However, you may use ftp:// or http:// protocols without a certificate.

1. First, as with all things Grid, you *must* have a valid proxy certificate to run globus-url-copy in certain protocols (gsiftp:// and https://, as noted above). If you are using ftp:// or http:// protocols, security is *not* mandatory and you may skip the rest of this table.

If you do not have a certificate, you must [obtain one](#).

If you are doing this for testing in your own environment, the [SimpleCA](#) provided with the Globus Toolkit should suffice.

If not, you must contact the Virtual Organization (VO) with which you are associated to find out whom to ask for a certificate.

One common source is the [DOE Science Grid CA](#)¹, although you must confirm whether or not the resources you wish to access will accept their certificates.

Instructions for proper installation of the certificate should be provided from the source of the certificate.

Please note when your certificates expire; they will need to be renewed or you may lose access to your resources.

2. Now that you have a certificate, you must generate a temporary proxy. Do this by running:

```
grid-proxy-init
```

Further documentation for **grid-proxy-init** can be found [here](#).

3. You are now ready to use **globus-url-copy**! See the following sections for syntax and command line options and other considerations.

Command syntax

The basic syntax for **globus-url-copy** is:

```
globus-url-copy [optional command line switches] Source_URL Destination_URL
```

where:

| | |
|----------------------------------|--|
| [optional command line switches] | See Command line options below for a list of available options. |
| <i>Source_URL</i> | Specifies the original URL of the file(s) to be copied. If this is a directory, all files within that directory will be copied. |
| <i>Destination_URL</i> | Specifies the URL where you want to copy the files. If you want to copy multiple files, this must be a directory. |



Note

Any url specifying a directory must end with */*.

URL prefixes

As of GT 3.2, we support the following URL prefixes:

- **file://** (on a local machine only)
- **ftp://**
- **gsiftp://**
- **http://**

¹ <http://www.doe grids.org/pages/cert-request.htm>

- **https://**

By default, **globus-url-copy** expects the same kind of host certificates that **globusrun** expects from gatekeepers.



Note

We do *not* provide an interactive client similar to the generic FTP client provided with Linux. See the [Interactive Clients](#) section below for information on an interactive client developed by NCSA/NMI/TeraGrid.

URL formats

URLs can be any valid URL as defined by RFC 1738 that have a protocol we support. In general, they have the following format: ***protocol://host:port/path***.



Note

If the path ends with a trailing / (i.e. /path/to/directory/) it will be considered to be a directory and all files in that directory will be moved. If you want a recursive directory move, you need to add the -r/-recurse switch described below.

Table 66. URL formats

| | |
|--|--|
| <code>gsiftp://myhost.mydomain.com:2812/data/foo.dat</code> | Fully specified. |
| <code>http://myhost.mydomain.com/mywebpage/default.html</code> | Port is not specified; therefore, GridFTP uses protocol default (in this case, 80). |
| <code>file:///foo.dat</code> | Host is not specified; therefore, GridFTP uses your local host. Port is not specified; therefore, GridFTP uses protocol default (in this case, 80). |
| <code>file:/foo.dat</code> | This is also valid but is not recommended because, while many servers (including ours) accept this format, it is <i>not</i> RFC conformant and is not recommended. |



Important

For GridFTP (`gsiftp://`) and FTP (`ftp://`), it is legal to specify a user name and password in the the URL as follows:

```
gsiftp://myname:[mypassword]@myhost.mydomain.com/foo.dat
```

If you are using GSI security, then you may specify the username (but you may *not* include the `:` or the password) and the grid-mapfile will be searched to see if that is a valid account mapping for your distinguished name (DN). If it is found, the server will setuid to that account. If not, it will fail. It will NOT fail back to your default account.

If you are using anonymous FTP, the username *must* be one of the usernames listed as a valid anonymous name and the password can be anything.

If you are using password authentication, you must specify both your username and password. **THIS IS HIGHLY DISCOURAGED, AS YOU ARE SENDING YOUR PASSWORD IN THE CLEAR ON THE NETWORK.** This is worse than no security; it is a false illusion of security.

Command line options

Informational Options

| | |
|------------------|--|
| -help -usage | Prints help. |
| -version | Prints the version of this program. |
| -versions | Prints the versions of all modules that this program uses. |
| -q -quiet | Suppresses all output for successful operation. |
| -vb -verbose | During the transfer, displays: <ul style="list-style-type: none"> • number of bytes transferred, • performance since the last update (currently every 5 seconds), and • average performance for the whole transfer. |
| -dbg -debugftp | Debugs FTP connections and prints the entire control channel protocol exchange to STDERR. Very useful for debugging. Please provide this any time you are requesting assistance with a globus-url-copy problem. |
| -list <url> | This option will display a directory listing for the given url. |

Utility Ease of Use Options

| | |
|------------------------------------|--|
| -a -ascii | Converts the file to/from ASCII format to/from local file format. |
| -b -binary | Does not apply any conversion to the files. This option is turned on by default. |
| -f <i>filename</i> | Reads a list of URL pairs from a filename. Each line should contain: <i>sourceURL destURL</i> Enclose URLs with spaces in double quotes ("). Blank lines and lines beginning with the hash sign (#) will be ignored. |
| -r -recurse | Copies files in subdirectories. |
| -notpt -no-third-party-transfers | Turns third-party transfers off (on by default). Site firewall and/or software configuration may prevent a connection between the two servers (a <i>third party transfer</i>). If this is the case, globus-url-copy will "relay" the data. It will do a GET from the source and a PUT to the destination. This obviously causes a performance penalty but will allow you to complete a transfer you otherwise could not do. |

Reliability Options

| | |
|-----------------|---------------------------------|
| -rst -restart | Restarts failed FTP operations. |
|-----------------|---------------------------------|

- rst-retries <retries> Specifies the maximum number of times to retry the operation before giving up on the transfer.
 Use 0 for infinite.
 The default value is 5.
- rst-interval <seconds> Specifies the interval in seconds to wait after a failure before retrying the transfer.
 Use 0 for an exponential backoff.
 The default value is 0.
- rst-timeout <seconds> Specifies the maximum time after a failure to keep retrying.
 Use 0 for no timeout.
 The default value is 0.

Performance Options

- tcp-bs <size> | -tcp-buffer-size <size> Specifies the size (in bytes) of the TCP buffer to be used by the underlying ftp data channels.

 **Important**

This is critical to good performance over the WAN.

[How do I pick a value?](#)

- p <parallelism> | -parallel <parallelism> Specifies the number of parallel data connections that should be used.

 **Note**

This is one of the most commonly used options.

[How do I pick a value?](#)

- bs <block size> | -block-size <block size> Specifies the size (in bytes) of the buffer to be used by the underlying transfer methods.

- pp **(New starting with GT 4.1.3)** Allows pipelining. GridFTP is a command response protocol. A client sends one command and then waits for a "Finished response" before sending another. Adding this overhead on a per-file basis for a large data set partitioned into many small files makes the performance suffer. Pipelining allows the client to have many outstanding, unacknowledged transfer commands at once. Instead of being forced to wait for the "Finished response" message, the client is free to send transfer commands at any time.

- mc *filename source_url* **(New starting with GT 4.2.1)** Transfers a single file to many destinations. File-name is a line-separated list of destination urls. For more information on this option, click [here](#).

Multicasting must be [enabled for use](#) on the server side.

Security Related Options

-s <subject> | -subject <subject> Specifies a subject to match with both the source and destination servers.



Note

Used when the server does not have access to the host certificate (usually when you are running the server as a user). See [the section called “If you run a GridFTP server by hand...”](#).

-ss <subject> | -source-subject <subject> Specifies a subject to match with the source server.



Note

Used when the server does not have access to the host certificate (usually when you are running the server as a user). See [the section called “If you run a GridFTP server by hand...”](#).

-ds <subject> | -dest-subject <subject> Specifies a subject to match with the destination server.



Note

Used when the server does not have access to the host certificate (usually when you are running the server as a user). See [the section called “If you run a GridFTP server by hand...”](#).

-nodcau | -no-data-channel-authentication Turns off data channel authentication for FTP transfers (the default is to authenticate the data channel).



Warning

We *do not* recommend this option, as it is a security risk.

-dcsafe | -data-channel-safe Sets data channel protection mode to SAFE.

Otherwise known as *integrity* or *checksumming*.

Guarantees that the data channel has not been altered, though a malicious party may have observed the data.



Warning

Rarely used as there is a substantial performance penalty.

-dcpriv | -data-channel-private Sets data channel protection mode to PRIVATE.

The data channel is encrypted and checksummed.

Guarantees that the data channel has not been altered and, if observed, it won't be understandable.

 **Warning**

VERY rarely used due to the VERY substantial performance penalty.

Default globus-url-copy usage

A **globus-url-copy** invocation using the **gsift** protocol with no options (i.e., using all the defaults) will perform a transfer with the following characteristics:

- binary
- stream mode (which implies no parallelism)
- host default TCP buffer size
- encrypted and checksummed control channel
- an authenticated data channel

MODES in GridFTP

GridFTP (as well as normal FTP) defines multiple wire protocols, or MODES, for the data channel.

Most normal FTP servers only implement *stream mode* (MODE S), i.e. the bytes flow in order over a single TCP connection. GridFTP defaults to this mode so that it is compatible with normal FTP servers.

However, GridFTP has another MODE, called Extended Block Mode, or *MODE E*. This mode sends the data over the data channel in blocks. Each block consists of 8 bits of flags, a 64 bit integer indicating the offset from the start of the transfer, and a 64 bit integer indicating the length of the block in bytes, followed by a payload of length bytes. Because the offset and length are provided, out of order arrival is acceptable, i.e. the 10th block could arrive before the 9th because you know explicitly where it belongs. This allows us to use multiple TCP channels. If you use the `-p | -parallelism` option, **globus-url-copy** automatically puts the servers into MODE E.



Note

Putting `-p 1` is not the same as no `-p` at all. Both will use a single stream, but the default will use stream mode and `-p 1` will use MODE E.

If you run a GridFTP server by hand...

If you run a GridFTP server by hand, you will need to explicitly specify the subject name to expect. The subject option provides **globus-url-copy** with a way to validate the remote servers with which it is communicating. Not only must the server trust **globus-url-copy**, but **globus-url-copy** must trust that it is talking to the correct server. The validation is done by comparing host DNs or subjects.

If the GridFTP server in question is running under a host certificate then the client assumes a subject name based on the server's canonical DNS name. However, if it was started under a user certificate, as is the case when a server is started by hand, then the expected subject name must be explicitly stated. This is done with the `-ss`, `-sd`, and `-s` options.

`-ss` Sets the `sourceURL` subject.

`-ds` Sets the `destURL` subject.

- s If you use this option alone, it will set both urls to be the same. You can see an example of this usage under the [Troubleshooting](#) section.



Note

This is an *unusual* use of the client. Most times you need to specify both URLs.

How do I choose a value?

How do I choose a value for the TCP buffer size (-tcp-bs) option?

The value you should pick for the TCP buffer size (-tcp-bs) depends on how fast you want to go (your bandwidth) and how far you are moving the data (as measured by the Round Trip Time (RTT) or the time it takes a packet to get to the destination and back).

To calculate the value for -tcp-bs, use the following formula (this assumes that Mega means 1000^2 rather than 1024^2, which is typical for bandwidth):

$$-tcp-bs = \text{bandwidth in Megabits per second (Mbs)} * \text{RTT in milliseconds (ms)} * 1000 / 8$$

As an example, if you are using fast ethernet (100 Mbs) and the RTT was 50 ms it would be:

$$-tcp-bs = 100 * 50 * 1000 / 8 = 625,000 \text{ bytes.}$$

So, how do you come up with values for bandwidth and RTT? To determine RTT, use either ping or traceroute. They both list RTT values.



Note

You must be on one end of the transfer and ping the other end. This means that if you are doing a third party transfer you have to run the ping or traceroute between the two server hosts, not from your client.

The bandwidth is a little trickier. Any point in the network can be the bottleneck, so you either need to talk with your network engineers to find out what the bottleneck link is or just assume that your host is the bottleneck and use the speed of your network interface card (NIC).



Note

The value you pick for -tcp-bs limits the top speed you can achieve. You will NOT get bandwidth any higher than what you used in the calculation (assuming the RTT is actually what you specified; it varies a little with network conditions). So, if for some reason you want to limit the bandwidth you get, you can do that by judicious choice of -tcp-bs values.

So where does this formula come from? Because it uses the bandwidth and the RTT (also known as the latency or delay) it is called the *bandwidth delay product*. The very simple explanation is this: TCP is a reliable protocol. It must save a copy of everything it sends out over the network until the other end acknowledges that it has been received.

As a simple example, if I can put one byte per second onto the network, and it takes 10 seconds for that byte to get there, and 10 seconds for the acknowledgment to get back (RTT = 20 seconds), then I would need at least 20 bytes of storage. Then, hopefully, by the time I am ready to send byte 21, I have received an acknowledgement for byte 1 and I can free that space in my buffer. If you want a more detailed explanation, try the following links on TCP tuning:

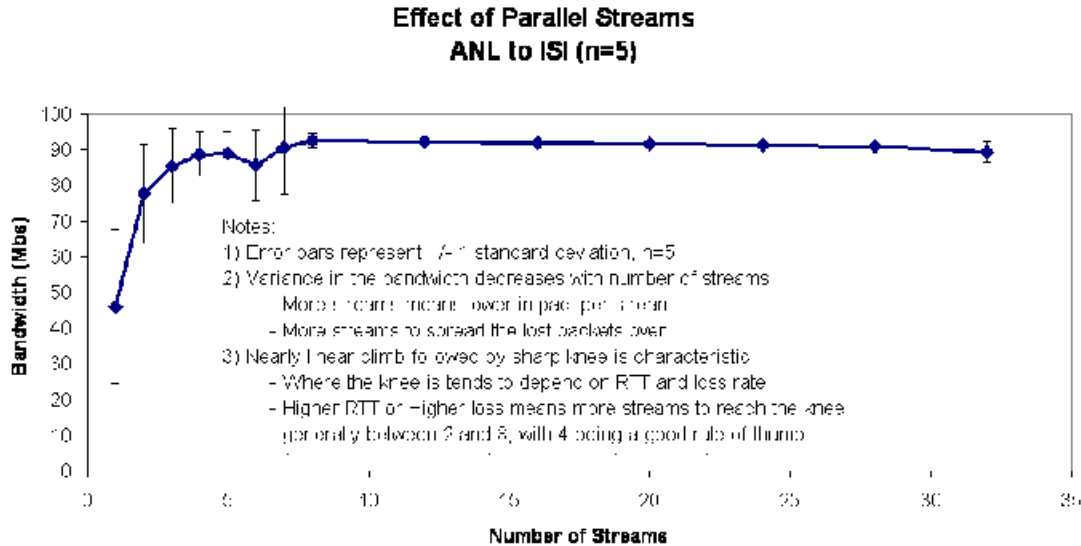
- http://www.psc.edu/networking/perf_tune.html

- <http://www.didc.lbl.gov/TCP-tuning/>
- <http://www.ncne.nlanr.net/research/tcp/>

How do I choose a value for the parallelism (-p) option?

For most instances, using 4 streams is a very good rule of thumb. Unfortunately, there is not a good formula for picking an exact answer. The shape of the graph shown here is very characteristic.

Figure 1. Effect of Parallel Streams in GridFTP



You get a strong, nearly linear, increase in bandwidth, then a sharp knee, after which additional streams have very little impact. Where this knee is depends on many things, but it is generally between 2 and 10 streams. Higher bandwidth, longer round trip times, and more congestion in the network (which you usually can only guess at based on how applications are behaving) will move the knee higher (more streams needed).

In practice, between 4 and 8 streams are usually sufficient. If things look really bad, try 16 and see how much difference that makes over 8. However, anything above 16, other than for academic interest, is basically wasting resources.

Limitations

There are no limitations for **globus-url-copy** in GT 4.2.1.

Interactive clients for GridFTP

The Globus Project does *not* provide an interactive client for GridFTP. Any normal FTP client will work with a GridFTP server, but it cannot take advantage of the advanced features of GridFTP. The interactive clients listed below take advantage of the advanced features of GridFTP.

There is no endorsement implied by their presence here. We make no assertion as to the quality or appropriateness of these tools, we simply provide this for your convenience. We will *not* answer questions, accept bugs, or in any way shape or form be responsible for these tools, although they should have mechanisms of their own for such things.

UberFTP was developed at the NCSA under the auspices of NMI and TeraGrid:

- NCSA Uerftp only download: <http://dims.ncsa.uiuc.edu/set/uberftp/download.html>
- UberFTP User's Guide: <http://dims.ncsa.uiuc.edu/set/uberftp/userdoc.html>

Name

globus-gridftp-server -- Configures the GridFTP Server

globus-gridftp-server

Tool description

globus-gridftp-server configures the GridFTP server using a config file and/or commandline options.



Note

Command line options and configuration file options may both be used, but the command line *overrides* the config file.

The configuration file for the GridFTP *server* is read from the following locations, in the given order. Only the first file found will be loaded:

- Path specified with the `-c <configfile>` command line option.
- `$GLOBUS_LOCATION/etc/gridftp.conf`
- `/etc/grid-security/gridftp.conf`

Options are one per line, with the format:

```
<option> <value>
```

If the value contains spaces, they should be enclosed in double-quotes ("). Flags or boolean options should only have a value of 0 or 1. Blank lines and lines beginning with # are ignored.

For example:

```
port 5000
allow_anonymous 1
anonymous_user bob
banner "Welcome!"
```

Developer notes

The Globus implementation of the GridFTP *server* draws on:

- three IETF RFCs:
 - RFC 959
 - RFC 2228
 - RFC 2389
- an IETF Draft: MLST-16
- the GridFTP protocol specification, which is Global Grid Forum (GGF) Standard GFD.020.

The command line tools and the *client* library completely hide the details of the protocol from the user and the developer. Unless you choose to use the control library, it is not necessary to have a detailed knowledge of the protocol.

Command syntax

The basic syntax for **globus-gridftp-server** is:

```
globus-gridftp-server [optional command line switches]
```

To use **globus-gridftp-server** with a config file, make sure to use the `-c <configfile>` option.

Command line options

The table below lists config file options, associated command line options (if available) and descriptions.



Note

Any boolean option can be negated on the command line by preceding the specified option with '-no-' or '-n'.
example: -no-cas or -nf.

Informational Options

| | |
|-------------------------------------|---|
| help <0 1>, -h, -help | Show usage information and exit. Default value: FALSE |
| version <0 1> , -v, -version | Show version information for the server and exit. Default value: FALSE |
| versions <0 1>, -v, -versions | Show version information for all loaded globus libraries and exit. Default value: FALSE |

Modes of Operation

| | |
|--------------------------------------|--|
| inetd <0 1>, -i, -inetd | Run under an inetd service. Default value: FALSE |
| daemon <0 1>, -s, -daemon | Run as a daemon. All connections will fork off a new process and setuid if allowed. See Section 4, “Running in daemon mode” for more information. Default value: TRUE |
| detach <0 1>, -S, -detach | Run as a background daemon detached from any controlling terminals. See Section 4, “Running in daemon mode” for more information. Default value: FALSE |
| exec <string> , -exec <string> | For statically compiled or non-GLOBUS_LOCATION standard binary locations, specify the full path of the server binary here. Only needed when run in daemon mode . Default value: not set |

`chdir <0|1>, -chdir` Change directory when the server starts. This will change directory to the dir specified by the `chdir_to` option.

Default value: TRUE

`chdir_to <string>, -chdir-to <string>` Directory to `chdir` to after starting. Will use `/` if not set.

Default value: not set

`fork <0|1>, -f, -fork` Server will fork for each new connection. Disabling this option is only recommended when debugging. Note that non-forked servers running as 'root' will only accept a single connection and then exit.

Default value: TRUE

`single <0|1>, -1, -single` Exit after a single connection.

Default value: FALSE

Authentication, Authorization, and Security Options

`auth_level <number>, -auth-level <number>`

- 0 = Disables all authorization checks.
- 1 = Authorize identity only.
- 2 = Authorize all file/resource accesses.

If not set, the GridFTP Server uses level 2 for front ends and level 1 for data nodes.

Default value: not set

`allow_from <string>, -allow-from <string>` Only allow connections from these source IP addresses. Specify a comma-separated list of IP address fragments. A match is any IP address that starts with the specified fragment. Example: '192.168.1.' will match and allow a connection from 192.168.1.45. Note that if this option is used, any address not specifically allowed will be denied.

Default value: not set

`deny_from <string>, -deny-from <string>` Deny connections from these source IP addresses. Specify a comma-separated list of IP address fragments. A match is any IP address that starts with the specified fragment. Example: '192.168.2.' will match and deny a connection from 192.168.2.45.

Default value: not set

`cas <0|1>, -cas` Enable Community Authorization Service (CAS) authorization. For complete instructions on setting up a GridFTP server to use CAS, click [here](#).

Default value: TRUE

`secure_ipc <0|1>, -si, -secure-ipc` Use GSI security on the IPC channel.

Default value: TRUE

| | |
|--|--|
| secure_ipc <0 1>, -si, -secure-ipc | Use GSI security on the IPC channel. Default value: TRUE |
| ipc_auth_mode <string>, -ia <string>, -ipc-auth- mode <string> | Set GSI authorization mode for the IPC connection. Options are one of the following: <ul style="list-style-type: none"> • none • host • self • subject:[subject] Default value: host |
| allow_anonym- ous <0 1>, -aa, -allow- anonymous | Allow cleartext anonymous access. If server is running as root, anonymous_user must also be set. Disables IPC security. Default value: FALSE |
| anonym- ous_names_al- lowed <string>, -an- onymous- names-allowed <string> | Comma-separated list of names to treat as anonymous users when allowing anonymous access. If not set, the default names of 'anonymous' and 'ftp' will be allowed. Use '*' to allow any user-name. Default value: not set |
| anonym- ous_user <string>, -an- onymous-user <string> | User to setuid to for an anonymous connection. Only applies when running as root. Default value: not set |
| anonym- ous_group <string>, -an- onymous-group <string> | Group to setgid to for an anonymous connection. If not set, the default group of anonymous_user will be used. Default value: not set |
| pw_file <string>, -password- file <string> | Enable cleartext access and authenticate users against this /etc/passwd formatted file. Default value: not set |
| connec- tions_max <number>, -connections- max <number> | Maximum concurrent connections allowed. Only applies when running in <u>daemon mode</u> . Unlimited if not set. Default value: not set |
| connec- tions_dis- abled <0 1>, Default value: FALSE | Disable all new connections. Does not affect ongoing connections. This must be set in the configuration file and then a SIGHUP issued to the server in order to reload the configuration. |

-connections-
disabled

Logging Options

log_level Log level. A comma-separated list of levels from the following:

<string>, -d
<string>,
-log-level
<string>

- ERROR
- WARN
- INFO
- DUMP
- ALL

For example:

```
globus-gridftp-server -d error,warn,info
```

You may also specify a numeric level of 1-255.

Default value: ERROR

log_module Indicates the globus_logging module that will be loaded. If not set, the default stdio module will be used and the logfile options (see next option) will apply.

<string>,
-log-module
<string>

Built-in modules are stdio and syslog. Log module options may be set by specifying module:opt1=val1:opt2=val2. Available options for the built-in modules are:

- interval - Indicates buffer flush interval. Default is 5 seconds. A 0 second flush interval will disable periodic flushing, and the buffer will only flush when it is full.
- buffer - Indicates buffer size. Default is 64k. A value of 0k will disable buffering and all messages will be written immediately.

Example:

```
-log-module stdio:buffer=4096:interval=10
```

Default value: not set

log_single Indicates the path of a single file to which you want to log all activity. If neither this option nor log_unique is set, logs will be written to stderr, unless the execution mode is detached, or inetd, in which case logging will be disabled.

<string>, -l
<string>,
-logfile
<string>

Default value: not set

log_unique Partial path to which gridftp.(pid).log will be appended to construct the log filename. Example:

<string>, -L
<string>, -logdir <string>

```
-L /var/log/gridftp/
```

will create a separate log (/var/log/gridftp/gridftp.xxxx.log) for each process (which is normally each new *client* session). If neither this option nor log_single is set, logs will be written to stderr, unless the execution mode is detached, or inetd, in which case logging will be disabled.

| | |
|--|--|
| | Default value: not set |
| log_transfer <string> , -Z <string> , -log-transfer <string> | Log NetLogger-style info for each transfer into this file. Default value: not set Example: DATE=20050520163008.306532 HOST=localhost PROG=globus-gridftp-server NL.EVNT=FTP_INFO START=20050520163008.305913 USER=ftp FILE=/etc/group BUF- FER=0 BLOCK=262144 NBYTES=542 VOLUME=/ STREAMS=1 STRIPES=1 DEST=[127.0.0.1] TYPE=RETR CODE=226 Time format is YYYYMMDDHHMMSS.UUUUUU (microsecs). <ul style="list-style-type: none"> • DATE: time the transfer completed. • START: time the transfer started. • HOST: hostname of the server. • USER: username on the host that transferred the file. • BUFFER: tcp buffer size (if 0 system defaults were used). • BLOCK: the size of the data block read from the disk and posted to the network. • NBYTES: the total number of bytes transferred. • VOLUME: the disk partition where the transfer file is stored. • STREAMS: the number of parallel TCP streams used in the transfer. • STRIPES: the number of stripes used on this end of the transfer. • DEST: the destination host. • TYPE: the transfer type, RETR is a send and STOR is a receive (ftp 959 commands). • CODE: the FTP rfc959 completion code of the transfer. 226 indicates success, 5xx or 4xx are failure codes. |
| log_filemode <string> , -log-filemode <string> | File access permissions of log files. Should be an octal number such as 0644 (the leading 0 is required). Default value: not set |
| disable_us- age_stats <0 1> , -dis- able-usage- stats | Disable transmission of per-transfer usage statistics. See the Usage Statistics ¹ section in the online documentation for more information. Default value: FALSE |
| us- age_stats_tar- get <string> , -usage-stats- | Comma-separated list of contact strings for usage statistics listeners. The format of <string> is host:port. Default value: usage-stats.globus.org:4810 |

¹ ../../Usage_Stats.html

target **Example:**
 <string>
 -usage-stats-target usage-stats.globus.org:4810,usage-stats.uc.teragrid.org

In this example, the usage statistics will be transmitted to the default Globus target (usage-stats.globus.org:4810) and another target (usage-stats.uc.teragrid.org:5920).

Single and Striped Remote Data Node Options

remote_nodes Comma-separated list of remote node contact strings. See [Remote data-nodes and striped operations](#) and [Separation of processes for higher security](#) for examples of using this option.
 <string>, -r
 <string>, -re-
mote-nodes Default value: not set
 <string>

data_node This server is a back end data node. See [Separation of processes for higher security](#) for an example of using this option.
 <0|1>, -dn,
 -data-node
 Default value: FALSE

stripe_blocks- Size in bytes of sequential data that each stripe will transfer.
ize <number>,
 -sbs <number> Default value: 1048576
 , -stripe-
blocksize
 <number>

stripe_layout Stripe layout. 1 = Partitioned, 2 = Blocked.
 <number>, -sl
 <number>,
 -stripe-lay-
out <number>
 Default value: 2

stripe_blocks- Do not allow client to override stripe blocksize with the **OPTS RETR** command.
ize_locked
 <0|1>,
 -stripe-block-
size-locked;
 Default value: FALSE

stripe_lay- Do not allow client to override stripe layout with the **OPTS RETR** command.
out_locked
 <0|1>,
 -stripe-lay-
out-locked
 Default value: FALSE

Disk Options

blocksize Size in bytes of data blocks to read from disk before posting to the network.
 <number>, -bs
 <number>,
 -blocksize
 <number>
 Default value: 262144

`sync_writes` `<0|1>`, `-sync-writes` Flush disk writes before sending a restart marker. This attempts to ensure that the range specified in the restart marker has actually been committed to disk. This option will probably impact performance and may result in different behavior on different storage systems. See the man page for `sync()` for more information.

Default value: FALSE

Network Options

`port` `<number>`, `-p` `<number>`, `-port` `<number>` Port on which a front end will listen for client control channel connections or on which a data node will listen for connections from a front end. If not set, a random port will be chosen and printed via the logging mechanism. See [Remote data-nodes and striped operations](#) and [Separation of processes for higher security](#) for examples of using this option.

Default value: not set

`control_interface` `<string>`, `-control-interface` `<string>` Hostname or IP address of the interface to listen for control connections on. If not set, will listen on all interfaces.

Default value: not set

`data_interface` `<string>`, `-data-interface` `<string>` Hostname or IP address of the interface to use for data connections. If not set will use the current control interface.

Default value: not set

`ipc_interface` `<string>`, `-ipc-interface` `<string>` Hostname or IP address of the interface to use for IPC connections. If not set, will listen on all interfaces.

Default value: not set

`hostname` `<string>`, `-hostname` `<string>` Effectively sets the above `control_interface`, `data_interface` and `ipc_interface` options.

Default value: not set

`ipc_port` `<number>`, `-ipc-port` `<number>` Port on which the front end will listen for data node connections.

Default value: not set

Timeouts

`control_preauth_timeout` `<number>`, `-control-preauth-timeout` `<number>` Time in seconds to allow a client to remain connected to the control channel without activity before authenticating.

Default value: 30

`control_idle_timeout` `<number>`; `-control-` Time in seconds to allow a client to remain connected to the control channel without activity.

Default value: 600

idle-timeout
<number>

ipc_idle_timeout Idle time in seconds before an unused IPC connection will close.
<number> ,
-ipc-idle- Default value: 600
timeout <num-
ber>

ipc_con- Time in seconds before cancelling an attempted IPC connection.
nect_timeout
<number> , Default value: 60
-ipc-connect-
timeout <num-
ber>

User Messages

banner Message that is displayed to the client before authentication.
<string> ,
-banner Default value: not set
<string>

banner_file Read banner message from this file.
<string> ,
-banner-file Default value: not set
<string>

banner_terse When this is set, the minimum allowed banner message will be displayed to unauthenticated
<0|1> , -ban- clients.
ner-terse
Default value: FALSE

login_msg Message that is displayed to the client after authentication.
<string> , -lo-
gin-msg Default value: not set
<string>

lo- Read login message from this file.
gin_msg_file
<string> , -lo- Default value: not set
gin-msg-file
<string>

Module Options

load_dsi_mod- Load this Data Storage Interface module. File and remote modules are defined by the server. If
ule <string> , not set, the file module is loaded, unless the remote option is specified, in which case the remote
-dsi <string> module is loaded. An additional configuration string can be passed to the DSI using the format
[module name]:[configuration string]. The format of the configuration string is
defined by the DSI being loaded.

Default value: not set

allowed_modules <string> Comma-separated list of ERET/ESTO modules to allow and, optionally, specify an alias for. Example:
 , -allowed-modules <string> -allowed-modules module1,alias2:module2,module3
 (module2 will be loaded when a client asks for alias2).
 Default value: not set

Other Options

configfile <string>, -c <string> Path to configuration file that should be loaded. Otherwise will attempt to load \$GLOBUS_LOCATION/etc/gridftp.conf and /etc/grid-security/gridftp.conf.
 Default value: not set

use_home_dirs <0|1>, -use-home-dirs Set the startup directory to the authenticated user's home dir.
 Default value: TRUE

debug <0|1>, -debug Set options that make the server easier to debug. Forces no-fork, no-chdir, and allows core dumps on bad signals instead of exiting cleanly. Not recommended for production servers. Note that non-forked servers running as root will only accept a single connection and then exit.
 Default value: FALSE

Limitations

For transfers using parallel data transport streams and for transfers using multiple computers at each end, the direction of the connection on the data channels must go from the sending to the receiving side. For more information about this limitations see <http://www.ogf.org/documents/GFD.20.pdf>.

Globus GridFTP server does not run on windows

RFT Commands

Name

rft -- Submit and monitor a 3rd party GridFTP transfer

rft

Tool description

Submits a transfer to the Reliable File Transfer Service and prints out the status of the transfer on the console.

Command syntax and options

```
rft [-h <host-ip of the container defaults to localhost>
-r <port, defaults to 8080>
-l <lifetime for the resource default 60mins>
-m <security mechanism. 'msg' for secure message or 'conv' for
  secure conversation and 'trans' for transport. Defaults to
  secure transport.>
-p <protection type, 'sig' signature and 'enc' encryption,
  defaults to signature >
-z <authorization mechanism can be self or host. default self>
-file <file to write EPR of created Reliable File Transfer Resource]>
-f <path to the file that contains list of transfers>
```

This is a sample transfer file that the command-line client will be able to parse. It can also be found in **\$GLOBUS_LOCATION/share/globus_wsrf_rft_client/** along with other samples for directory transfers and deletes (lines starting with # are comments):

```
This option when it is set to true means to perform transfer in binary
form, if it is set to false transfer is done in ASCII. Default is binary.
true
```

```
#Block size in bytes that is transferred. Default is 16000 bytes.
16000
```

```
#TCP Buffer size in bytes
```

```
#Specifies the size (in bytes) of the TCP buffer to be used by the underlying
ftp data channels. This is critical to good performance over the WAN. Use the
bandwidth-delay product as your buffer size.
```

```
16000
```

```
#Notpt (No thirdPartyTransfer): turns third-party transfers off is this option
is set to false (on if set to true).
```

```
Site firewall and/or software configuration may prevent a connection
between the two servers (a third party transfer). If this is the case,
RFT will "relay" the data. It will do a GET from the source and a PUT to
the destination. This obviously causes a performance penalty, but will allow
you to complete a transfer you otherwise could not do.
```

```
false
```

```
#Number of parallel streams: Specifies the number of parallel data connections
that should be used.
```

1

#Data Channel Authentication (DCAU): Turns off data channel authentication for FTP transfers is set to false.(the default is true to authenticate the data channel).

true

Concurrency of the request: Number of files that you want to transfer at any given point. Default is set to one.

1

#Grid Subject name of the source gridftp server. This is used for Authorization purposes. If the source gridftp server is running with host credentials you can specify "n /DC=org/DC=doegrids/OU=People/CN=Ravi Madduri 134710

#Grid Subject name of the destination gridftp server. This is used for Authorization purposes. If the destination gridftp server is running with host credentials you can specify "null" here. By default Host authorization is done. /DC=org/DC=doegrids/OU=People/CN=Ravi Madduri 134710

#Transfer all or none of the transfers: This option if set to true will make RFT to clean up (delete) all the transfers that have been done already if one of the transfers fails.

false

#Maximum number of retries: This is number of times RFT retries a transfer failed with a non-zero exit code.

10

#Source/Dest URL Pairs: gsiftp urls of source followed by destination.

If directory is to be recursively transferred the source gsiftp url and destination gsiftp url should end with "/". Currently RFT supports Directory - Directory, File - Directory, File - File transfers. There can be more URL pairs and all of them use the same options as above for performing the transfer.

gsiftp://localhost:5678/tmp/rftTest.tmp

gsiftp://localhost:5678/tmp/rftTest_Done.tmp

Limitations

This command line client is very simple and does not do any intelligent parsing of various command line options or of the options in the sample transfer file. It works fine if used in the way documented here. For more information on all these options please refer to the [documentation of globus-url-copy](#). Also, please note that the maximum number of transfers the command-line client can process before running out of memory is ~21K with the default JVM heap size, which was 64M in our tests. Please look at [Performance Reports](#)¹ for more details.

¹ ../rft_scalability_3_9_4.doc

Name

globus-crft -- Command-line client to transfer files using RFT

globus-crft

Tool description

This distribution contains a client to the RFT service written in C. RFT is the reliable transfer server. It allows clients to submit URL transfer requests to a persistent service which will perform the transfers on behalf of the client.

Options

| | |
|---|--|
| <code>-a --all-or-none <on off></code> | Enable all or none transfer: default off. |
| <code>-con --concurrent <int></code> | The number of simultaneous transfers. |
| <code>-C --cancel</code> | Cancel a transfer. |
| <code>-c --create</code> | Create a new RFT service. |
| <code>-del --delete</code> | Delete a URL. |
| <code>-ds --destination-subject <subject></code> | The expected domain name of the destination GridFTP server. |
| <code>-d --destroy</code> | Destroy the server. If used with <code>-monitor</code> , wait until completion and then destroy. |
| <code>-D --done</code> | Return the current status of the transfer in the exit code: <ul style="list-style-type: none">• 0=Done• 1=Active• 2=Pending• 3=Cancelled• 4=Failed |
| <code>-ef --epr-file <path></code> | Path to the EPR file. If used with <code>--create</code> the EPR is written to this location. In all other cases the EPR is read from this location. |
| <code>-ez --easy</code> | Create, submit, and wait for the transfer to complete. The job is started with some standard options. |
| <code>-e --factory <contact></code> | The endpoint to contact when creating a server. Used with <code>--create</code> . |
| <code>-f --transfer-file <path></code> | A path to a file that contains the source destination URL pairs. |
| <code>-gS --getStatusSet <int> <int></code> | Get the status of all the transfer requests in the range. |
| <code>-g --getStatus <source url></code> | Get the status of the given source url. |

`-h` | `--help` Print usage information.

FIXME - finish converting to variable list:

`-ms` | `--message-security` <[sig] | [conv] | [trans]>
 Security mechanism. 'msg' for secure message, 'conv' for secure conversation, 'trans' for transport. The default is trans.

`-m` | `--monitor` Wait for the service to complete, and receive status updates.

`-os` | `--getOverallStatus` Get the overall status.

`-p` | `--protection` <[sig] | [enc]>
 Protection type. 'sig' for signature, 'enc' for encryption. The default is 'sig'.

`-P` | `--parallel` <int> The number of parallel sockets to use with each transfer.

`-q` | `--quiet` Write no output.

`-rs` | `--getRequestStatus` Get the request status.

`-r` | `--retries` Number of retries

`-S` | `--subject` <subject> The expected domain name of both the source and destination GridFTP servers.

`-ss` | `--source-subject` <subject>
 The expected domain name of the source GridFTP server.

`-s` | `--submit` Start the RFT service

`-tb` | `--tcp-bs` <int> The TCP buffer size to use with each transfer.

`-ttl` | `--termination-time` <int>
 Set the lifetime of the service.

`-v` | `--version` Print version information.

`-vb` | `--verbose` Display much more output.

`-xi` | `--xml-input` <path> Read the request description from the given xml description.

`-xo` | `--xml-output` <path> Write the request description to the given file location in xml format.

`-z` | `--authz` <[self] | [host] | [id <subject>]>
 Authorization. 'self', 'host', or 'id <DN>'.

Limitations

No limitations exist with this command line tool.

Name

rft-delete -- Command-line client to delete files using RFT

rft-delete

Tool description

This command-line tool is used to submit a list of files to be deleted.

Command and options

```
rft-delete [-h <host-ip of the container default localhost>
-r <port, defaults to 8080>
-l <lifetime for the resource default 60mins>
-m <security mechanism. 'msg' for secure message or 'conv' for
  secure conversation and 'trans' for transport. Defaults to
  secure transport.>
-p <protection type, 'sig' signature and 'enc' encryption,
  defaults to signature >
-z <authorization mechanism can be self or host. default self>
-file <file to write EPR of created Reliable File Transfer Resource]>
-f <path to the file that contains list of transfers>
```

This is a sample file that the command line client will be able to parse, and it can also be found in **\$GLOBUS_LOCATION/share/globus_wsrf_rft_client/** along with other samples for directory transfers and deletes (lines starting with # are comments):

```
# Subject name (defaults to host subject)
  /DC=org/DC=doegrids/OU=People/CN=Ravi Madduri 134710
  gsiftp://localhost:5678/tmp/rftTest_Done.tmp
  gsiftp://localhost:5678/tmp/rftTest_Done1.tmp
```

Limitations

No limitations exist with this command line tool.

Replica Location Service (RLS) Commands

Name

globus-rls-admin -- RLS administration tool

globus-rls-admin

Tool description

Performs administrative operations on an RLS server.

Synopsis

-A/-a/-C option value/-c option/-d/-e/-p/-q/-s/-t timeout/-u/-v [rli] [pattern] [server]

Options

Table 67. Options for globus-rls-admin

| | |
|------------------------|---|
| -A | <p>Adds rli to the list of <i>RLI</i> servers updated by an <i>LRC</i> server using <i>Bloom filters</i>.</p> <p><i>Note:</i> Partitions are not supported with Bloom filters. The LRC server maintains one Bloom filter for all <i>LFNs</i> in its database, which is sent to all RLI servers configured to receive Bloom filter updates with this option.</p> |
| -a | <p>Adds rli and optionally pattern to the list of RLI servers that the LRC server sends updates to (using a list of LFNs).</p> <p>If pattern is specified, then only LFNs matching it will be sent to rli.</p> <p>If rli is added with no patterns, then it is sent all updates. Pattern matching is done using standard Unix file globbing.</p> |
| -C option value | <p>Sets server option to value.</p> <p><i>Important:</i> This does <i>not</i> update the configuration file. The next time the server is restarted, the configuration change will be <i>lost</i>.</p> |
| -c option | <p>Retrieves the configuration value for the specified option from the server.</p> <p>If option is set to all, then all options are retrieved.</p> |
| -d | <p>Removes rli and pattern from the list of RLI servers that the LRC server sends updates to.</p> <p>If pattern is not specified, then all entries for rli are removed.</p> <p><i>Note:</i> If all patterns are removed separately, then rli is sent all updates. To stop any updates from being sent to rli, do <i>not</i> specify pattern.</p> |
| -e | <p>Clears the LRC database. Removes all lfn, <i>pfn</i> mappings.</p> |
| -p | <p>Verifies that the server is responding.</p> |
| -q | <p>Causes the RLS server to exit.</p> |
| -S | <p>Shows statistics and other information gathered by the RLS server.</p> <p>This is intended to be input into GRIS.</p> |
| -s | <p>Shows the list of RLI servers and patterns being sent updates by the LRC server.</p> <p>If rli or pattern are not specified, they are considered wildcards.</p> |
| -t timeout | <p>Sets timeout (in seconds) for RLS server requests.</p> <p>The default value is 30.</p> |
| -u | <p>Causes the LRC server to immediately start full soft state updates to any RLI servers previously added with the -a option.</p> |
| -v | <p>Shows the version and exits.</p> |

Name

globus-rls-cli -- RLS client tool

globus-rls-cli

Tool description

Provides a command line interface to some of the functions supported by RLS. It also supports an interactive interface (if *command* is not specified). In interactive mode, double quotes may be used to encode an argument that contains white space.

Synopsis

command [*-c*] [*-h*] [*-l reslimit*] [*-s*] [*-t timeout*] [*-u*] [*command*] *rls-server*

Options

The client command tool uses **getopt** for command line parsing.

Note: Some versions will continue scanning for options (works that begin with a hyphen) for the entire command line, which makes it impossible to specify negative integer or floating point value for an *attribute*. The workaround for this problem is to tell **getopt()** that there are no more options by including 2 hyphens. For example, to specify the value **-2** you must enter **-- -2**.

Table 68. Options for globus-rls-cli

| | |
|--------------------|---|
| -c | Sets "clearvalues" flag when deleting an attribute (will remove any attribute value records when an attribute is deleted). |
| -h | Shows usage. |
| -l reslimit | Sets an incremental limit on the number of results returned by a wildcard query at a time. Note that <i>all results will be returned</i> by the client. This parameter only limits the number of results <i>incrementally retrieved</i> by the client during a single internal communication call. For instance, if the wildcard query produces 1000 results and the reslimit is set to 100, the client will internally make 10 calls to the server. From the user's perspective the client will simply return all 1000 results. Zero means no limit. |
| -s | Uses SQL style wildcards (% and _). |
| -t timeout | Sets timeout (in seconds) for RLS server requests. The default is 30 seconds. |
| -u | Uses Unix style wildcards (* and ?). |
| -v | Shows version. |

Commands

Table 69. Commands for globus-rls-cli

| | |
|--|--|
| add <lfn> <pfn> | Adds <i>pfn</i> to mappings of <i>lfn</i> in an <i>LRC</i> catalog. |
| attribute add <object> <attr> <obj-type> <attr-type> | Adds an attribute to an object, where <i>object</i> should be the lfn or pfn name. <i>obj-type</i> should be one of lfn or pfn. <i>attr-type</i> should be one of date, float, int, or string. If <value> is of type date then it should be in the form "YYYY-MM-DD HH:MM:DD". |
| attribute bulk add <object> <attr> <obj-type> | Bulk adds attribute values. |
| attribute bulk delete <object> <attr> <obj-type> | Bulk deletes attributes. |
| attribute bulk query <attr> <obj-type> <object> | Bulk queries attributes. |
| attribute define <attr> <obj-type> <attr-type> | Defines a new attribute. |
| attribute delete <object> <attr> <obj-type> | Removes <i>attribute</i> from <i>object</i> . |
| attribute modify <object> <attr> <obj-type> <attr-type> | Modifies the value of an attribute. |
| attribute query <object> <attr> <obj-type> | Retrieves the value of the specified attribute for object . |
| attribute search <attr> <obj-type> <operator> <attr-type> | Searches for objects which have the specified attribute matching <i>operator</i> and <i>value</i> . <i>operator</i> should be one of =, !=, >, >=, <, or <=. |
| attribute show <attr> <obj-type> | Shows an attribute definition. If <i>attr</i> is a hyphen (-) then all attributes are shown. |
| attribute undefine <attr> <obj-type> | Deletes an attribute definition. Will return an error if any objects possess this attribute. |
| bulk add <lfn> <pfn> [<lfn> <pfn>] | Bulk adds lfn, pfn mappings. |
| bulk create <lfn> <pfn> [<lfn> <pfn>] | Bulk creates lfn, pfn mappings. |
| bulk delete <lfn> <pfn> [<lfn> <pfn>] | Bulk deletes lfn, pfn mappings. |
| bulk query lrc lfn [<lfn> ...] | Bulk queries the LRC for lfns. |
| bulk query lrc pfn [<pfn> ...] | Bulk queries the LRC for pfns. |
| bulk query rli lfn [<lfn> ...] | Bulk queries the <i>RLI</i> for lfns. |
| create <lfn> <pfn> | Creates a new <i>lfn</i> , <i>pfn</i> mapping in an LRC catalog. |
| delete <lfn> <pfn> | Deletes a <i>lfn</i> , <i>pfn</i> mapping from an LRC catalog. |
| exit | Exits the interactive session. |
| help | Prints a help message. |
| query lrc lfn <lfn> | Queries an LRC server for mappings of <i>lfn</i> . |
| query lrc pfn <pfn> | Queries an LRC server for mappings to <i>pfn</i> . |
| query rli lfn <lfn> | Queries an RLI server for mappings of <i>lfn</i> . |

| | |
|---|--|
| query wildcard lrc lfn <lfn-pattern> | Performs a wildcarded query of an LRC server for mappings of <i>lfn-pattern</i> . Patterns use the standard Unix wildcard characters: an asterisk (*) matches 0 or more characters, and a question mark (?) matches any single character. |
| query wildcard lrc pfn <pfn-pattern> | Queries an LRC server for mappings to <i>pfn-pattern</i> . Patterns use the standard Unix wildcard characters: an asterisk (*) matches 0 or more characters, and a question mark (?) matches any single character. |
| query wildcard rli lfn <lfn-pattern> | Queries an RLI server for mappings of <i>lfn-pattern</i> . Patterns use the standard Unix wildcard characters: an asterisk (*) matches 0 or more characters, and a question mark (?) matches any single character. |
| set reslimit <limit> | <p>Sets an incremental limit on the number of results returned by a wildcard query at a time.</p> <p>Note that <i>all results will be returned</i> by the client. This parameter only limits the number of results <i>incrementally retrieved</i> by the client during a single internal communication call. For instance, if the wildcard query produces 1000 results and the reslimit is set to 100, the client will internally make 10 calls to the server. From the user's perspective the client will simply return all 1000 results.</p> |
| set timeout <timeout> | <p>Sets the timeout (in seconds) on calls to the RLS server.</p> <p>The default value is 30.</p> |
| version | Shows the version and exits. |

Name

globus-rls-server -- RLS server tool

globus-rls-server

Tool description

The RLS server (**globus-rls-server**) can be configured as either one or both of the following:

- *Location Replica Catalog (LRC)* server, which manages *Logical FileName (LFN)* to *Physical FileName (PFN)* mappings in a database. *Note*: If **globus-rls-server** is configured as an LRC server, the *RLI* servers that it sends updates to should be added to the database using **globus-rls-admin**.
- *Replica Location Index (RLI)* server, which manages mappings of LFNs to LRC servers.

Clients wishing to locate one or more physical filenames associated with a logical filename should first contact an RLI server, which will return a list of LRCs that may know about the LFN. The LRC servers are then contacted in turn to find the physical filenames.

Note: RLI information may be out of date, so clients should be prepared to get a negative response when contacting an LRC (or no response at all if the LRC server is unavailable).

Synopsis

```
[ -B lrc_update_bf ] [ -b maxbackoff ] [ -C rlscertfile ] [ -c conffile ] [ -d ] [ -e rli_expire_int ] [ -F lrc_update_factor ] [ -f maxfreethreads ] [ -I true/false ] [ -i idletimeout ] [ -K rlskeyfile ] [ -L loglevel ] [ -l true/false ] [ -M maxconnections ] [ -m maxthreads ] [ -N ] [ -o lrc_buffer_time ] [ -p pidfiledir ] [ -r true/false ] [ -S rli_expire_stale ] [ -s starthreads ] [ -t timeout ] [ -U myurl ] [ -u lrc_update_ll ] [ -v ]
```

LRC to RLI Updates

Two methods exist for LRC servers to inform RLI servers of their LFNs.

- By default, the LFNs are sent from the LRC to the RLI. This can be time consuming if the number of LFNs is large, but it does give the RLI an exact list of the LFNs known to the LRC, and it allows wildcard searching of the RLI.
- Alternatively, *Bloom filters* may be sent, which are highly compressed summaries of the LFNs. However, they do not allow wildcard searching and will generate more "false positives" when querying an RLI.

Please see below for more on Bloom filters.

globus-rls-admin can be used to manage the list of RLIs that an LRC server updates. This includes partitioning LFNs among multiple RLI servers.

A soft state algorithm is used in both update modes: periodically the LRC server sends its state (LFN information) to the RLI servers it updates. The RLI servers add these LFNs to their indexes or update timestamps if the LFNs were already known. RLI servers expire information about LFN, LRC mappings if they haven't been updated for a period longer than the soft state update interval.

The following options in the configuration file control the soft state algorithm when an LRC updates an RLI by sending LFNs:

- **rli_expire_int** (seconds)
- **rli_expire_stale** (seconds)
- **lrc_update_ll** (seconds)
- **lrc_update_bf** (seconds)

Updates to an LRC (new LFNs or deleted LFNs) normally don't propagate to RLI servers until the next soft state update (controlled by options **lrc_update_ll** and **lrc_update_bf**).

However, by enabling "immediate update" mode (set **lrc_update_immediate** to **true**), an LRC will send updates to an RLI within **lrc_buffer_time** seconds.

If updates are done with LFN lists then only the LFNs that have been added or deleted to the LRC are sent. If Bloom filters are used, then the entire Bloom filter is sent.

When immediate updates are enabled, the interval between soft state updates is multiplied by **lrc_update_factor** as long as no updates have failed (LRC and RLI are considered to be in sync). This can greatly reduce the number of soft state updates an LRC needs to send to an RLI.

Incremental updates are buffered by the LRC server until either 200 updates have accumulated (when LFN lists are used), or **lrc_buffer_time** seconds have passed since the last update.

Bloom filter updates

A Bloom filter is an array of bits. Each LFN is hashed multiple times and the corresponding bits in the Bloom filter are set.

Querying an RLI to verify if an LFN exists is done by performing the same hashes and checking if the bits in the filter are on. If not, then the LFN is known not to exist. If they're all on, then all that's known is that the LFN probably exists.

The size of the Bloom filter (as a multiple of the number of LFNs) and the number of hash functions control the false positive rate. The default values of 10 and 3 give a false positive rate of approximately 1%.

The advantage of Bloom filters is their efficiency. For example, if the LRC has 1,000,000 LFNs in its database, with an average length of 20 bytes, then 20,000,000 bytes must be sent to an RLI during a soft state update (assuming no partitioning). The RLI server must perform 1,000,000 updates to its database to create new LFN, LRC mappings or update timestamps on existing entries. With Bloom filters only 1,250,000 bytes are sent (10 x 1,000,000 bits / 8), and there are no database operations on the RLI (Bloom filters are maintained entirely in memory). A comparison of the time to perform a 1,000,000 LFN update: it took 20 minutes sending all the LFNs and less than 1 second using a Bloom filter. However as noted before, Bloom filters do *not* support wild card searches of an RLI.

Note: An LRC server can update some RLIs with Bloom filters and others with LFNs. However, an RLI server can only be updated using one method.

The following options in the [Configuration](#) file control Bloom filter updates:

- **rli_bloomfilter** true|false
- **rli_bloomfilter_dir** none|default|pathname
- **lrc_bloomfilter_numhash** N
- **lrc_bloomfilter_ratio** N

- `irc_update_bf` seconds

Log Messages

globus-rls-server uses syslog to log errors and other information (facility **LOG_DAEMON**) when it's running in normal (daemon) mode.

If the **-d** option (debug) is specified, then log messages are written to stdout.

Signals

The server will reread its configuration file if it receives a **HUP** signal. It will wait for all current requests to complete and shut down cleanly if sent any of the following signals: **INT**, **QUIT** or **TERM**.

Options (globus-rls-server)

The following table describes the command line options available for globus-rls-server:

Table 70. Options for globus-rls-server

| | |
|-----------------------------|--|
| -b maxbackoff | Maximum time (in seconds) that globus-rls-server will attempt to reopen the socket it listens on after an I/O error. |
| -C rls-certfile | Name of the X.509 certificate file that identifies the server; sets environment variable X509_USER_CERT . |
| -c conffile | Name of the configuration file for the server. The default is \$GLOBUS_LOCATION/etc/globus-rls-server.conf if the environment variable GLOBUS_LOCATION is set; else, /usr/local/etc/globus-rls-server.conf . |
| -d | Enables debugging. The server will not detach from the controlling terminal, and log messages will be written to stdout rather than syslog. For additional logging verbosity set the loglevel (see the -L option) to higher values. |
| -e rli_expire_int | Interval (seconds) at which an RLI server should expire stale entries. |
| -F irc_update_factor | If irc_update_immediate mode is on, and the LRC server is in sync with an RLI server (an LRC and RLI are synced if there have been no failed updates since the last full soft state update), then the interval between RLI updates for this server (irc_update_ll) is multiplied by irc_update_factor . |
| -f maxfreethreads | Maximum number of idle threads the server will leave running. Excess threads are terminated. |
| -I true false | Turns LRC to RLI immediate update mode on (true) or off (false). The default value is false . |
| -i idletimeout | Seconds after which idle client connections are timed out. |
| -K rls-keyfile | Name of the X.509 key file. Sets environment variable X509_USER_KEY . |
| -L loglevel | Sets the log level. By default this is 0 , which means only errors will be logged. Higher values mean more verbose logging. |
| -l true false | Configures whether the server is an LRC server. The default is false . |
| -M maxconnections | Maximum number of active connections. It should be small enough to prevent the server from running out of open file descriptors. The default value is 100 . |
| -m maxthreads | Maximum number of threads server will start up to support simultaneous requests. |
| -N | Disables authentication checking. This option is intended for debugging. Clients should use the URL RLSN://host to disable authentication on the client side. |
| -o irc_buffer_time | LRC to RLI updates are buffered until either the buffer is full or this much time (in seconds) has elapsed since the last update. The default value is 30 . |
| -p pidfiledir | Directory where PID files should be written. |

| | |
|----------------------------|--|
| -r | Configures whether the server is an RLI server. The default value is false . |
| -S rli_expire_stale | Interval (in seconds) after which entries in the RLI database are considered stale (presumably because they were deleted in the LRC). Stale entries are not returned in queries. |
| -s startthreads | Number of threads to start up initially. |
| -t timeout | Timeout (in seconds) for calls to other RLS servers (in other words, for LRC calls to send an update to an RLI). A value of 0 disables timeouts. The default value is 30 . |
| -U myurl | URL for this server. |
| -u lrc_update_ll | Interval (in seconds) between lfn-list LRC to RLI updates. |
| -v | Shows version and exits. |

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 71. globus-repicalocation-createmappings 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-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 72. 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 73. 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-replication-defineattributes -- This tool is used to define attributes.

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

Table 74. globus-replication-defineattributes Options

| | |
|---|---|
| { name object-type value-type }+ | A listing of attribute name, associated object-type, and value-type. |
| input-file | A file containing the listing of attribute name, associated object-type, and value-type. |
| - | 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 75. globus-repicalocation-undefineattributes Options

| | |
|------------------------------|--|
| { name object-type }+ | A listing of attribute name and associated object-type. |
| input-file | A file containing the listing of attribute name and associated object-type. |
| - | 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 76. globus-repicalocation-addattributes Options

| | |
|---|---|
| { key name object-type value-type value }+ | A listing of key, attribute name, associated object-type, value-type, and value. |
| input-file | A file containing the listing of key, attribute name, associated object-type, value-type, and value. |
| - | 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 77. globus-replication-modifyattributes Options

| | |
|---|---|
| { key name object-type value-type value }+ | A listing of key, attribute name, associated object-type, value-type, and value. |
| input-file | A file containing the listing of key, attribute name, associated object-type, value-type, and value. |
| - | 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 78. globus-repicalocation-removeattributes Options

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

DataRep Commands

The Batch Replicator 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 referred 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 79. 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 80. 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 81. 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 82. 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 83. 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 84. 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' |

Replication Client Commands

The Replication Client consists of a single command-line tool and an API. The client accepts different *commands* (e.g., get, put, register,...) and calls RLS and GridFTP Server(s) to perform the operations.

Name

globus-replication-client -- Performs several intuitive data replication operations.

globus-replication-client

Tool description

The command-line client supports intuitive data replication operations such as get (locate a replica and retrieve it), put (transfer local data to a remote location and register it as a replica), copy (locate a replica and copy the data to another location), replicate (locate a replica, copy the data to another location, and register the new location as a replica), delete (delete a specific replica), and register (register an existing data file in the replica catalog).

Command syntax

globus-replication-client [options] command

Table 85. Options

| | |
|--|--|
| -a --ascii | Use ASCII type transfer. Default: off. |
| -b --binary | Use binary type transfer. Default: on. |
| -h --help | Display help. |
| -nodcaul --no-data-channel-authentication | Turns off data channel authentication for FTP transfers. Default: authenticate the data channel. |
| -p --parallel <size> | Specifies the number of parallel data connections that should be used. |
| -r --registry <url> | Specifies the replica name service that should be used. Example: rls://localhost. |
| -s --subject <subject> | Specifies a subject name to use with both the source and destination servers. |
| -tcpbs --tcp-buffer-size <size> | Specifies the size (in bytes) of the TCP buffer to be used by the underlying ftp data channels. |
| -v --verbose | Displays verbose output. |

Table 86. Commands And Arguments

| | |
|--|---|
| copy <name> <url> | Copies a replica. The new copy (at the given url) will NOT be registered in the RLS. |
| delete <name> <url> | Deletes the specific instance of the replicated data. |
| get <name> <file> | Locates the replica and retrieves a copy to the local file. |
| put {<file> <url>} <name> <url> | Transfers data from a local file or remote url to a destination url and registers it in the RLS under the given name. |
| register <name> <url> | Registers data specified by url under a given name in the RLS. |
| replicate <name> <url> | Locates a replica, transfers the data to the specified location, and registers it in the RLS. |

WS MDS Commands

Name

`mds-servicegroup-add` -- Registering grid resources to aggregating MDS services such as the Index, Archive and Trigger services

`mds-servicegroup-add`

Tool description

mds-servicegroup-add creates a set of registrations to a WS-ServiceGroup and periodically renews those registrations. It is intended primarily for registering grid resources to aggregating MDS services such as the Index and Trigger services.

The tool can be deployed at the aggregating service, at resource services, or at any other location.

This allows registrations to be configured by the administrator of the aggregating service, or by the administrator of resources, by a third party, or by some combination of those.

Registrations are defined in an XML configuration file, which is documented here: [Chapter 3, Registering Aggregator Sources](#).

For an example using an Index Service, see [Simple usage for the Index Service](#).

And remember to note the section on [Limitations](#).

Command syntax

The basic syntax for **mds-servicegroup-add** is:

```
mds-servicegroup-add [options] config.xml
```

where:

| | |
|--|---|
| <code>-s ht-tp(s)://host:port/service-group-address</code> | A URL to the service group against which the <code>mds-servicegroup-add</code> request will be executed. This command line argument is an optional argument, it is only necessary that this URL argument be specified in the case that there are no suitable target service group EPRs present in the configuration file . Any end point references found in the configuration file will automatically override the EPR specified by this argument on the command-line. If this argument is not specified and no suitable service group EPR is present in the configuration file, the target EPR defaults to the <code>DefaultIndexService</code> on the local host using the default TLS port of 8443. |
| <code>-o outputFile</code> | If this argument is specified, mds-servicegroup-add will write the EPRs of all successfully created service group entries from the target resource to this file. This file can then be used as input to the mds-set-multiple-termination-time command. |
| <code>-q seconds</code> | By default, mds-servicegroup-add will continue to run, refreshing the lifetimes for the service group entry resources it creates. Use this option to cause mds-servicegroup-add to terminate itself after the specified number of seconds has elapsed. This can be helpful when using long-lifetime registrations or when updating entry lifetimes via a different mechanism. |
| <code>-a</code> | By default, mds-servicegroup-add will attempt to make an authenticated connection to each service group. This option is used to specify anonymous connections (and to prevent mds-servicegroup-add from failing if you don't have a valid Grid credential). |
| <code>-z auth_type</code> | Specify an authorization type: |

| | |
|-------------------------|--|
| | <p><code>self</code> Fail if the server's identity is different from the user's identity.</p> <p><code>host</code> Fail if the server does not have a valid server certificate.</p> <p><code>none</code> Continue regardless of the server's identity.</p> |
| <code>config.xml</code> | <p>Path to the registration configuration file (see Chapter 3, Registering Aggregator Sources).</p> <p>The Globus Toolkit distribution includes an example configuration file: <code>\$GLOBUS_LOCATION/etc/globus_wsrif_mds_aggregator/example-aggregator-registration.xml</code>.</p> |

The common `java` client options are also supported.

Registering a resource manually

Prerequisites

You need the following before you register a resource with an Index Service:



Note

Beginning with GT 4.0.1, the CAS, RFT and GRAM4 services are automatically registered with the default Index Service.

- Have a working Index Service, as documented in the [Index Service System Administrator's Guide](#).
- Know the EPR to the resource.
- Know the EPR to the Index Service. This can be seen in the container output at startup of the container on the index host, and will look something like this: `https://myhost:8443/wsrif/services/DefaultIndexService`

Simple usage for the Index Service

The simplest way to register resources to an index is to:

1. Edit the example configuration file (`$GLOBUS_LOCATION/etc/globus_wsrif_mds_aggregator/example-aggregator-registration.xml`), replacing the EPRs in that file with the EPRs for your resources
2. Run `mds-servicegroup-add` to perform the registrations specified in that file.

For example, to register to the `DefaultIndexService` with a modified `example-aggregator-registration.xml` file, you could run a command similar to the following:

```
$GLOBUS_LOCATION/bin/mds-servicegroup-add -s \
https://127.0.0.1:8443/wsrif/services/DefaultIndexService \
$GLOBUS_LOCATION/etc/globus_wsrif_mds_aggregator/example-aggregator-registratio
```

Limitations

It may be necessary for the tool to continue to run in order for the registrations that it maintains to be kept alive, as registrations will otherwise time out.

Name

`mds-set-multiple-termination-time` -- Administering the termination time of grid resources created by aggregating MDS services such as the Index and Trigger services

`mds-set-multiple-termination-time`

Tool description

mds-set-multiple-termination-time sets the termination time of multiple service group entries. It is intended primarily for working with groups of service group entry resources created by aggregating MDS services such as the Index and Trigger services.

The tool can be deployed at the aggregating service, at resource services, or at any other location.

This allows the lifetime of registrations to be configured by the administrator of the aggregating service, or by the administrator of resources, by a third party, or by some combination of those.

Command syntax

The basic syntax for **mds-set-multiple-termination-time** is:

`mds-set-multiple-termination-time [options]`

where:

| | |
|-----------------------------|---|
| <code>-i inputFile</code> | file containing an XML array of Endpoint References, such as one output by the mds-servicegroup-add command when used with the <code>-o</code> option. |
| <code>-w delay</code> | integer wait delay in seconds that will be added to the current time at the remote resource to generate the resource termination time. If not specified the termination time by defaults is set to the current time at the remote resource. |
| <code>-n date string</code> | ISO-8601 formatted date string representing an exact date and time, e.g. 2016-06-28T01:06:430Z If not specified the termination time by default is set to the current time at the remote resource. |

The [common java client options](#) are also supported.

GRAM4 Commands

Name

globusrun-ws -- Official job submission client for GRAM4

```
globusrun-ws -submit [-batch] [-quiet] [-no-cleanup] [-streaming] [-streaming-out filename] [-streaming-err filename] [-host-authz] [-self-authz] [-subject-authz subject name] [-private] [-http-timeout milliseconds] [-debug] [-allow-ipv6] [-passive] [-nodcau] [[-factory-epr-file filename] [[-factory contact] | [-factory-type type]]] [[-submission-id uuid] | [-submission-id-file filename]] [-submission-id-output-file filename] [-job-epr-output-file filename] [-job-delegate] [-staging-delegate] [-job-credential-file filename] [-staging-credential-file filename] [-transfer-credential-file filename] [-termination [+HH:MMmm/dd/yyyy HH:MM] ] [[-job-description-file filename] | [-job-command [--] program arg ...]]
globusrun-ws -validate -job-description-file filename
globusrun-ws -monitor -job-epr-file filename [-quiet] [-no-cleanup] [-streaming] [-streaming-out filename] [-streaming-err filename] [-host-authz] [-self-authz] [-subject-authz subject name] [-private] [-http-timeout milliseconds] [-debug] [-allow-ipv6] [-passive] [-nodcau]
globusrun-ws -status -job-epr-file filename [-host-authz] [-self-authz] [-subject-authz subject name] [-private] [-http-timeout milliseconds] [-debug]
globusrun-ws -kill -job-epr-file filename [-host-authz] [-self-authz] [-subject-authz subject name] [-private] [-http-timeout milliseconds] [-debug]
globusrun-ws -help
globusrun-ws -usage [-submit] [-validate] [-monitor] [-status] [-kill]
globusrun-ws -version(s)
```

Description

globusrun-ws (GRAM4 client) is a program for submitting and managing jobs to a local or remote job host. GRAM4 provides secure job submission to many types of *job scheduler* for users who have the right to access a job hosting resource in a Grid environment. All GRAM4 submission options are supported transparently through the embedded request document input. globusrun-ws offers additional features to fetch job output files incrementally during the run as well as to automatically delegate credentials needed for certain optional GRAM4 features. Online and batch submission modes are supported with reattachment (recovery) for jobs whether they were started with this client or another GRAM4 client application.

Command options

Quiet mode

A variety of protocol status messages, warning messages, and output data may be printed to standard output and error under multiple command modes. The *quiet mode* suppresses all but fatal standard error messages in order to have clean outputs for use in scripting or with the *streaming output mode* where application output is retrieved and output.

-q, -quiet If supplied, all non-fatal status and protocol-related messages are suppressed.

Debug mode

-dbg, -debug If supplied, all soap messages and ftp control messages will be displayed on stderr.

Protocol Options

Service authorization

Usually, secure communication includes mutual authentication. In addition to the service authorizing the client for the requested operation(s), an authorization decision is made by the client to determine whether the remote service is the one intended.

- | | |
|--|--|
| -host, -host-authz | The GSI "host authorization" rule is used to verify that the service is using a host credential appropriate for the underlying service address information. This is the default. |
| -self, -self-authz | The GSI "self authorization" rule is used to verify that the service is using a (proxy) credential derived from the same identity as the client's. |
| -subject, -subject-authz <u>subject name</u> | The service must be using a credential with the exact subject name provided by this option. |

Security Protocol

The client uses secure transport for all https endpoints and secure message for http. Secure conversation is currently unsupported.

- | | |
|--------------|---|
| -p, -private | If supplied, privacy-protection is enabled between globusrun-ws and GRAM4 or GridFTP services. It is a fatal error to select privacy protection if it is not available due to build options or other security settings. Note: Currently only supported with https endpoints. |
|--------------|---|

Timeouts

- | | |
|--|---|
| -T, -http-timeout <u>milli-seconds</u> | Set timeout for HTTP socket, in milliseconds, for all Web services interactions. The default value is 120000 (2 minutes). |
|--|---|

Signal handling

- | | |
|-----------------|--|
| -n, -no-cleanup | If supplied, the default behavior of trapping interrupts (SIG_INTR) and cancelling the job is disabled. Instead, the interrupt simply causes the tool to exit without affecting the ManagedJob resource. |
|-----------------|--|

Submit options

- | | |
|---------|--|
| -submit | The -submit command submits (or <i>resubmits</i>) a job to a job host using an <u>XML-based job description</u> document. The -submit command can submit jobs in one of three output modes: batch, interactive, or interactive-streaming. |
|---------|--|

Output Mode

The user can select several tool behaviors following submission. In *batch mode*, the tool prints the resulting ManagedJob EPR as the sole standard output (unless in *quiet mode*) and exits. In *interactive mode*, the tool keeps running in order to monitor job status. Interactive mode is qualitatively equivalent to a batch-mode submission immediately followed a second invocation of globusrun-ws using the -monitor command. In interactive mode, an optional *streaming mode* where job output files are fetched and output from globusrun-ws.

- b, -batch If supplied, the batch mode is enabled. The default is interactive mode. The tool prints the resulting ManagedJob EPR as the sole standard output (unless in quiet mode) and exits.
- s, -streaming The standard output and standard error files of the job are monitored and data is written to the corresponding output of globusrun-ws. The standard output will contain ONLY job output data, while the standard error may be a mixture of job error output as well as globusrun-ws messages, unless the *quiet mode* is also enabled.
- Streaming output depends on the ability to access job outputs via GridFTP. If -streaming mode is selected and the *job description* does not already specify output file redirection for the job host, then globusrun-ws adds unique output file name redirections and automatic cleanup directives to the job description.
- If you are using -batch mode, but intend to use -streaming with -monitor, you may want to still include -streaming. -streaming always introduces a 'CleanUp Hold' state which ensures that all the data is streamed before the files are destroyed. If you do use -streaming with -batch, you **must** come back with -monitor so the hold can be released.
- This option implies -staging-delegate if the stdout and stderr entries are not specified in the job description.
- so, -stdout-file filename append stdout out stream to the specified file instead of to stdout.
- se, -stderr-file filename append stderr out stream to the specified file instead of to stderr.

Streaming Options

Streaming makes use of GridFTP client calls to retrieve user data. The following options apply to such transfers.

- ipv6, -allow-ipv6 Allow streaming transfers to use IPV6.
- passive Force streaming transfers to use MODE S to allow for passive mode transfers. (Useful if you're behind a firewall, but expensive because there is no connection caching).
- nodcau Disable data channel authentication on streaming transfers

Factory information

Addressing information for the ManagedJobFactory target of this submission must be provided. If neither option is specified, and no EPR is supplied in the job description, then "-factory localhost -factory-type fork" is assumed.

- Ff, -factory-epr-file filename If supplied, this option causes the EPR for the ManagedJobFactory to be read from the given file. This EPR is used as the service endpoint for submission of the job.
- F, -factory contact If supplied, this option causes an EPR to be constructed using ad-hoc methods that depend on GT implementation details. For interoperability to other implementations of GRAM4_, the -factory-epr-file option should be used instead.

[protocol://][{hostname|hostaddr}][:port][/service]

Default values form the following contact information if not overridden:

https://localhost:8443/wsrp/services/ManagedJobFactoryService

-Ft, -factory-type type In the absence of `-factory-epr-file`, this option refines the behavior of the `-factory` option to select a specific type of scheduler. The default is "Fork" for single jobs and "Multi" for *multijobs*.

Job description

A description of the job to be submitted must be provided with the `-submit` command, either using the GRAM4 XML description syntax or a simpler Unix command and argument list.

-f, -job-description-file filename If supplied, this option causes the job description to be read from the given file. This description is modified according to the other options and passed in the GRAM4 submission messages. The root element of this file must be 'job' for a single job or 'multiJob' for a multijob.

-c, -job-command [--] prog [arg ...] If supplied, this option take all remaining globusrun-ws arguments as its arguments; therefore it must appear last among globusrun-ws options. This option causes globusrun-ws to generate a simple job description with the named program and arguments.

Submission ID

A submission ID may be used in the GRAM4 protocol for robust reliability in the face of message faults or other transient errors to ensure that at most one instance of a job is executed, i.e. to prevent accidental duplication of jobs under rare circumstances with client retry on failure. The globusrun-ws tool always uses this feature, requiring either a submission ID to be passed in as input or a new unique ID to be created by the tool itself. If a new ID is created, it should be captured by the user who wishes to exploit this reliability interface. The ID in use, whether created or passed as input, will be written to the optional output file when provided, as well as to the standard error output unless the *quiet mode* is in effect.

If a user is unsure whether a job was submitted successfully, he should resubmit using the same ID as was used for the previous attempt.

-I, -submission-id ID If supplied, this option causes the job to be submitted using the given ID in the reliability protocol.

-If, -submission-id-file filename If supplied, this option causes the ID to be read from the given file. It is an error to use both mechanisms to provide an input ID.

-Io, -submission-id-output-file file-name If supplied, the ID in use is written to the given file, whether this ID was provided by the user or given by one of the above input options.

Job EPR output

A successful submission will create a new ManagedJob resource with its own unique EPR for messaging. The globusrun-ws tool will output this EPR to a file when requested and as the sole standard output when running in batch mode. When running in streaming output mode, it is possible that the EPR will not be output and the user's only recourse is to submit again with the same submission ID and job request in order to reattach to the existing job.

-o, -job-epr-output-file filename If supplied, the created ManagedJob EPR will be written to the given file following successful submission. The file will not be written if the submission fails.

Delegation

The job description supports the optional identification of delegated credentials for use by the GRAM4 services. These features are passed through globusrun-ws without modification. However, globusrun-ws can also perform delegation

and construct these optional request elements before submitting it to the service. The only delegation performed by default (if an endpoint does not already exist) is the multijob level jobCredential.

- J, -job-delegate If supplied AND the job description does not already provide a jobCredential element, globusrun-ws will delegate the client credential to GRAM4 and introduce the corresponding element to the submission input.
- S, -staging-delegate If supplied AND the job description does include staging or cleanup directives AND the job description does not already provide the necessary stagingCredential or transferCredential element(s), globusrun-ws will delegate the client credential to GRAM4 and RFT, and introduce the corresponding elements to the submission input.

This option is implied by -streaming
- Jf, -job-credential-file filename: If supplied AND the job description does not already provide a jobCredential element, globusrun-ws will copy the supplied epr into the job description. This should be an epr returned from the DelegationFactoryService intended for use by the job (or, in the case of a multijob, for authenticating to the subjobs).

note: for multijob descriptions, only the top level jobCredential will be copied into.
- Sf, -staging-credential-file filename: If supplied AND the job description does not already provide a stagingCredential element, globusrun-ws will copy the supplied epr into the job description. This should be an epr returned from the DelegationFactoryService intended for use with the RFT service associated with the ManagedJobService.

note: this option is ignored for multijobs.
- Tf, -transfer-credential-file filename: If supplied, globusrun-ws will copy the epr into each of the stage in, stage out, and cleanup elements that do not already contain a transferCredential element. This should be an epr returned from the DelegationFactoryService intended for use by RFT to authenticate with the target gridftp server.

note: this option is ignored for multijobs.

Lifetime

The ManagedJob resource supports lifetime management in the form of a scheduled destruction. The default lifetime requested by the client is infinite, subject to server policies.

- term, -termination mm/dd/yyyy Set an absolute termination time.
HH:MM
- term, -termination +HH:MM Set a termination time relative to the successful creation of the job. The default is +24:00

Validate options

- validate The -validate command checks the job description for syntax errors and a subset of semantic errors without making any service requests.

Job description

-f, -job-description-file filename This option causes the job description to be read from the given file. This description is checked for validity.

Monitor options

-monitor The -monitor command attaches to an existing job in interactive or interactive-streaming output modes.

Job

Addressing information for the ManagedJob target of this command must be provided.

-j, -job-epr-file filename If supplied, this option causes the EPR for the ManagedJob to be read from the given file. This EPR is used as the endpoint for service requests.

Output mode

In the default *interactive mode*, the tool keeps running in order to monitor job status. In the optional *interactive-streaming mode*, the job output files are fetched and output from globusrun-ws as well.

-s, -streaming See Output mode under Submit Options above for details on streaming.

Status options

-status The -status command reports the current state of the job and exits. See the [External States of the Managed Job Services](#) section of the developer guid for information on valid job states.

See the Job options for the -monitor command.

Kill options

-kill The -kill command requests the immediate cancellation of the job and exits.

Help options

-help Outputs an overview of the commands and features of the command.

Usage options

-usage Outputs brief usage information for the command.

Version options

-version Outputs version information for the command.

Job Handling

For every job that globusrun-ws delegates a credential, globusrun will augment the user's job description, adding annotations that will later tell globusrun-ws to destroy the credential after the job has been destroyed. Below are 2 job annotation examples. globusrun-ws only delegated the job cred...

```
<extensions>
<globusrunAnnotation>
<automaticJobDelegation>true</automaticJobDelegation>
<automaticStagingDelegation>false</automaticStagingDelegation>
<automaticStageInDelegation>false</automaticStageInDelegation>
<automaticStageOutDelegation>false</automaticStageOutDelegation>
<automaticCleanUpDelegation>false</automaticCleanUpDelegation>
</globusrunAnnotation>
</extensions>
```

globusrun-ws delegated the job, staging and stage in cred...

```
<extensions>
<globusrunAnnotation>
<automaticJobDelegation>true</automaticJobDelegation>
<automaticStagingDelegation>true</automaticStagingDelegation>
<automaticStageInDelegation>true</automaticStageInDelegation>
<automaticStageOutDelegation>false</automaticStageOutDelegation>
<automaticCleanUpDelegation>false</automaticCleanUpDelegation>
</globusrunAnnotation>
</extensions>
```

Environment

X509_USER_PROXY Overrides the default selection of user credentials when using GSI security.

Exit Codes

The client returns negative error codes for client errors, 0 for success, and positive error codes from the submitted job (where possible)

GridWay Commands

Name

Job and Array Job submission Command -- job submission utility for the GridWay system

```
gws submit <-t template> [-n tasks] [-h] [-v] [-o] [-s start] [-i increment] [-d  
"id1 id2 ..."]
```

Description

Submit a job or an array job (if the number of tasks is defined) to gwd

Command options

| | |
|-------------------|---|
| -h | Prints help. |
| -t <template> | The template file describing the job. |
| -n <tasks> | Submit an array job with the given number of tasks. All the jobs in the array will use the same template. |
| -s <start> | Start value for custom param in array jobs. Default 0. |
| -i <increment> | Increment value for custom param in array jobs. Each task has associated the value $PARAM=start + increment * TASK_ID$, and $MAX_PARAM = start+increment*(tasks-1)$. Default 1. |
| -d <"id1 id2..."> | Job dependencies. Submit the job on hold state, and release it once jobs with id1,id2,.. have successfully finished. |
| -v | Print to stdout the job ids returned by gwd. |
| -o | Hold job on submission. |
| -p <priority> | Initial priority for the job. |

Name

DAG Job submission Command -- dag job submission utility for the GridWay system

```
gwdag [-h] [-d] <DAG description file>
```

Description

Submit a dag job to gwd

Command options

-h Prints help.

-d Writes to STDOUT a DOT description for the specified DAG job.

Name

Job Monitoring Command -- report a snapshot of the current jobs

```
gwps [-h] [-u user] [-r host] [-A AID] [-s job_state] [-o output_format] [-c  
delay] [-n] [job_id]
```

Description

Prints information about all the jobs in the GridWay system (default)

Command options

-h Prints help.

-u user Monitor only jobs owned by user.

-r host Monitor only jobs executed in host.

-A AID Monitor only jobs part of the array AID.

-s job_state Monitor only jobs in states matching that of job_state.

-o output_format Formats output information, allowing the selection of which fields to display.

-c <delay> This will cause gwps to print job information every <delay> seconds continuously (similar to top command).

-n Do not print the header.

job_id Only monitor this job_id.

Output field description

Table 87. Field options

| FIELD NAME | FIELD OPTION | DESCRIPTION | |
|------------|--------------|--|--|
| USER | u | owner of this job | |
| JID | J | job unique identification assigned by the Gridway system | |
| AID | i | array unique identification, only relevant for array jobs | |
| TID | i | task identification, ranges from 0 to TOTAL_TASKS -1, only relevant for array jobs | |
| FP | p | fixed priority of the job | |
| TYPE | y | type of job (simple, multiple or mpi) | |
| NP | n | number of processors | |
| DM | s | dispatch Manager state, one of: pend, hold, prol, prew, wrap, epil, canl, stop, migr, done, fail | |
| EM | e | execution Manager state (Globus state): pend, susp, actv, fail, done | |
| RWS | f | flags: | |
| | | R | times this job has been restarted |
| | | W | number of processes waiting for this job |
| | | S | re-schedule flag |
| START | t T | the time the job entered the system | |
| END | t T | the time the job reached a final state (fail or done) | |
| EXEC | t T | total execution time, includes suspension time in the remote queue system | |
| XFER | t T | total file transfer time, includes stage-in and stage-out phases | |
| EXIT | x | job exit code | |
| TEMPLATE | j | filename of the job template used for this job | |
| HOST | h | hostname where the job is being executed | |

Name

Job History Command -- shows history of a job

```
gwhistory [-h] [-n] <job_id>
```

Description

Prints information about the execution history of a job

Command options

-h Prints help.

-n Do not print the header lines

job_id Job identification as provided by gwps.

Output field description

Table 88. Field information

| NAME | DESCRIPTION |
|---------|--|
| HID | host unique identification assigned by the Gridway system. |
| START | the time the job start its execution on this host. |
| END | the time the job left this host, because it finished or it was migrated. |
| PROLOG | total prolog (file stage-in phase) time. |
| WRAPPER | total wrapper (execution phase) time. |
| EPILOG | total epilog (file stage-out phase) time. |
| MIGR | total migration time. |
| REASON | the reason why the job left this host. |
| QUEUE | name of the queue. |
| HOST | FQDN of the host. |

Name

Host Monitoring Command -- shows hosts information

```
gwhost [-h] [-c delay] [-nf] [-m job_id] [host_id]
```

Description

Prints information about all the hosts in the GridWay system (default)

Command options

-h Prints help.

-c <delay> This will cause gwhost to print job information every <delay> seconds continuously (similar to top command)

-n Do not print the header.

-f Full format.

-m <job_id> Prints hosts matching the requirements of a given job.

host_id Only monitor this host_id, also prints queue information.

Output field description

Table 89. Field information

| FIELD | DESCRIPTION |
|-----------|---|
| HID | host unique identification assigned by the Gridway system |
| PRIO | priority assigned to the host |
| OS | operating system |
| ARCH | architecture |
| MHZ | CPU speed in MHZ |
| %CPU | free CPU ratio |
| MEM(F/T) | system memory: F = Free, T = Total |
| DISK(F/T) | secondary storage: F = Free, T = Total |
| N(U/F/T) | number of slots: U = used by GridWay, F = free, T = total |
| LRMS | local resource management system, the jobmanager name |
| HOSTNAME | FQDN of this host |

Table 90. Queue field information

| FIELD | DESCRIPTION |
|--------------|----------------------------|
| QUEUENAME | name of this queue |
| SL(F/T) | slots: F = Free, T = Total |
| WALLT | queue wall time |
| CPUT | queue cpu time |
| COUNT | queue count number |
| MAXR | max. running jobs |
| MAXQ | max. queued jobs |
| STATUS | queue status |
| DISPATCH | queue dispatch type |
| PRIORITY | queue priority |

Name

Job Control Command -- controls job execution

```
gkill [-h] [-a] [-k | -t | -o | -s | -r | -l | -9] <job_id [job_id2 ...] | -A  
array_id>
```

Description

Sends a signal to a job or array job

Command options

- h Prints help.
 - a Asynchronous signal, only relevant for KILL and STOP.
 - k Kill (default, if no signal specified).
 - t Stop job.
 - r Resume job.
 - o Hold job.
 - l Release job.
 - s Re-schedule job.
 - 9 Hard kill, removes the job from the system without synchronizing remote job execution or cleaning remote host.
- job_id [job_id2 ...] Job identification as provided by gwps. You can specify a blank space separated list of job ids.
- A <array_id> Array identification as provided by gwps.

Name

Job Synchronization Command -- synchronize a job

```
gwwait [-h] [-a] [-v] [-k] <job_id ...| -A array_id>
```

Description

Waits for a job or array job

Command options

-h Prints help.

-a Any, returns when the first job of the list or array finishes.

-v Prints job exit code.

-k Keep jobs, they remain in fail or done states in the GridWay system. By default, jobs are killed and their resources freed.

-A <array_id> Array identification as provided by gwps.

job_id ... Job ids list (blank space separated).

Name

User Monitoring Command -- monitors users in GridWay

```
gwuser [-h] [-n]
```

Description

Prints information about users in the GridWay system

Command options

-h Prints help.

-n Do not print the header.

Output field description

Table 91. Field information

| FIELD | DESCRIPTION |
|-------|---|
| UID | user unique identification assigned by the Gridway system |
| NAME | name of this user |
| JOBS | number of Jobs in the GridWay system |
| RUN | number of running jobs |
| IDLE | idle time, (time with JOBS = 0) |
| EM | execution manager drivers loaded for this user |
| TM | transfer manager drivers loaded for this user |
| PID | process identification of driver processes |

Name

Accounting Command -- prints accounting information

```
gwacct [-h] [-n] [<-d n | -w n | -m n | -t s>] <-u user|-r host>
```

Description

Prints usage statistics per user or resource. Note: accounting statistics are updated once a job is killed.

Command options

-h Prints help.

-n Do not print the header.

<-d n | -w n | -m n | -t s> Take into account jobs submitted after certain date, specified in number of days (-d), weeks (-w), months (-m) or an epoch (-t).

-u user Print usage statistics for user.

-r hostname Print usage statistics for host.

Output field description

Table 92. Field information

| FIELD | DESCRIPTION |
|-----------|--|
| HOST/USER | host/user usage summary for this user/host |
| XFR | total transfer time on this host (for this user) |
| EXE | total execution time on this host (for this user), without suspension time |
| SUSP | total suspension (queue) time on this host (for this user) |
| TOTS | total executions on this host (for this user). Termination reasons: <ul style="list-style-type: none">• SUCC success• ERR error• KILL kill• USER user requested• SUSP suspension timeout• DISC discovery timeout• SELF self migration• PERF performance degradation• S/R stop/resume |

Name

JSDL To GridWay Job Template Parser Command -- parser to translate JSDL file into GridWay Job Template file

```
jsdl2gw [-h] input_jsdl [output_gwjt]
```

Description

Converts a jsdl document into a gridway job template. If no output file is defined, it defaults to the standard output. This enables the use of pipes with gws submit in the following fashion:

```
jsdl2gw jsdl-job.xml | gws submit
```

Command options

-h Prints help.

input_jsdl Reads the jsdl document from the input_jsdl

output_gwjt Stores the GridWay Job Template specification in the output_gwjt.jt file

Glossary

B

Bloom filter Compression scheme used by the Replica Location Service (RLS) that is intended to reduce the size of soft state updates between Local Replica Catalogs (LRCs) and Replica Location Index (RLI) servers. A Bloom filter is a bit map that summarizes the contents of a Local Replica Catalog (LRC). An LRC constructs the bit map by applying a series of hash functions to each logical name registered in the LRC and setting the corresponding bits.

C

Certificate Authority (CA) An entity that issues certificates. [fixme - flesh out]

certificate A public key plus information about the certificate owner bound together by the digital signature of a CA. In the case of a CA certificate, the certificate is self signed, i.e. it was signed using its own private key.

client A process that sends commands and receives responses. Note that in GridFTP, the client may or may not take part in the actual movement of data.

E

extended block mode (MODE E) MODE E is a critical GridFTP components because it allows for out of order reception of data. This in turn, means we can send the data down multiple paths and do not need to worry if one of the paths is slower than the others and the data arrives out of order. This enables parallelism and striping within GridFTP. In MODE E, a series of “blocks” are sent over the data channel. Each block consists of:

- an 8 bit flag field,
- a 64 bit field indicating the offset in the transfer,
- and a 64 bit field indicating the length of the payload,
- followed by length bytes of payload.

Note that since the offset and length are included in the block, out of order reception is possible, as long as the receiving side can handle it, either via something like a seek on a file, or via some application level buffering and ordering logic that will wait for the out of order blocks.

F

flavor Pre-OGSI Globus description term that uniquely encompasses Machine Architecture, OS, Compiler and other attributes into a single term, for example: gcc32dbgpthr for a threaded Linux debug distribution.

G

grid map file A file containing entries mapping certificate subjects to local user names. This file can also serve as a access control list for GSI enabled services and is typically found in `/etc/grid-security/grid-mapfile`. For more information see the Gridmap section [here](#).

J

job description Term used to describe a GRAM4 job for GT4.

job scheduler See the term [scheduler](#)⁴.

L

Local Replica Catalog (LRC) Stores mappings between logical names for data items and the target names (often the physical locations) of replicas of those items. Clients query the LRC to discover replicas associated with a logical name. Also may associate attributes with logical or target names. Each LRC periodically sends information about its logical name mappings to one or more RLIs.

See also [RLI](#)⁶.

logical file name A unique identifier for the contents of a file.

M

multijob A job that is itself composed of several executable jobs; these are processed by the MMJS subjob.

See also [MMJS subjob](#)¹⁰.

P

physical file name The address or the location of a copy of a file on a storage system.

proxy certificate A short lived certificate issued using a EEC. A proxy certificate typically has the same effective subject as the EEC that issued it and can thus be used in its place. GSI uses proxy certificates for single sign on and delegation of rights to other entities.

For more information about types of proxy certificates and their compatibility in different versions of GT, see <http://dev.globus.org/wiki/Security/ProxyCertTypes>.

⁴ #scheduler

⁶ #rli

¹⁰ #mmjs-subjob

R

- Replica Location Index (RLI)** Collects information about the logical name mappings stored in one or more Local Replica Catalogs (LRCs) and answers queries about those mappings. Each RLI periodically receives updates from one or more LRCs that summarize their contents.
- RLS attribute** Descriptive information that may be associated with a logical or target name mapping registered in a Local Replica Catalog (LRC). Clients can query the LRC to discover logical names or target names that have specified RLS attributes.

S

- server** A process that receives commands and sends responses to those commands. Since it is a server or service, and it receives commands, it must be listening on a port somewhere to receive the commands. Both FTP and GridFTP have IANA registered ports. For FTP it is port 21, for GridFTP it is port 2811. This is normally handled via `inetd` or `xinetd` on Unix variants. However, it is also possible to implement a daemon that listens on the specified port. This is described more fully in the Architecture section of the GridFTP Developer's Guide.
- SOAP** SOAP provides a standard, extensible, composable framework for packaging and exchanging XML messages between a service provider and a service requester. SOAP is independent of the underlying transport protocol, but is most commonly carried on HTTP. See the [SOAP specifications](#)¹³ for details.
- stream mode (MODE S)** The only mode normally implemented for FTP is MODE S. This is simply sending each byte, one after another over the socket in order, with no application level framing of any kind. This is the default and is what a standard FTP server will use. This is also the default for GridFTP.

T

- third party transfers** In the simplest terms, a third party transfer moves a file between two GridFTP servers.
- The following is a more detailed, programmatic description.
- In a third party transfer, there are three entities involved. The client, who will only orchestrate, but not actually take place in the data transfer, and two servers one of which will be sending data to the other. This scenario is common in Grid applications where you may wish to stage data from a data store somewhere to a super-computer you have reserved. The commands are quite similar to the client/server transfer. However, now the client must establish two control channels, one to each server. He will then choose one to listen, and send it the PASV command. When it responds with the IP/port it is listening on, the client will send that IP/port as part of the PORT command to the other server. This will cause the second server to connect to the first server, rather than the client. To initiate the actual movement of the data, the client then sends the RETR "filename" command to the server that will read from disk and write to the network (the "sending" server) and will send

¹³ <http://www.w3.org/TR/soap/>

the STOR “filename” command to the other server which will read from the network and write to the disk (the “receiving” server).
See Also [client/server transfer](#).

transport-level security

Uses transport-level security (TLS) mechanisms.

W

Web Services Addressing (WSA)

The WS-Addressing specification defines transport-neutral mechanisms to address web services and messages. Specifically, it defines XML elements to identify web service endpoints and to secure end-to-end endpoint identification in messages. See the [W3C WS Addressing Working Group](#)¹⁴ for details.

Web Services Description Language (WSDL)

WSDL is an XML document for describing Web services. Standardized binding conventions define how to use WSDL in conjunction with SOAP and other messaging substrates. WSDL interfaces can be compiled to generate proxy code that constructs messages and manages communications on behalf of the client application. The proxy automatically maps the XML message structures into native language objects that can be directly manipulated by the application. The proxy frees the developer from having to understand and manipulate XML. See the [WSDL 1.1 specification](#)¹⁵ for details.

Web Services Resource Framework (WSRF)

Web Services Resource Framework (WSRF) is a specification that extends web services for grid applications by giving them the ability to retain state information while at the same time retaining statelessness (using resources). The combination of a web service and a resource is referred to as a WS-Resource. WSRF is a collection of different specifications that manage WS-Resources.

This framework comprises mechanisms to describe views on the state (WS-ResourceProperties), to support management of the state through properties associated with the Web service (WS-ResourceLifetime), to describe how these mechanisms are extensible to groups of Web services (WS-ServiceGroup), and to deal with faults (WS-BaseFaults).

For more information, go to: <http://www.globus.org/wsrf/> and [OASIS Web Services Notification \(WSRF\) TC](#)¹⁹.

X

XML

Extensible Markup Language (XML) is standard, flexible, and extensible data format used for web services. See the [W3C XML site](#)²⁰ for details.

¹⁴ <http://www.w3.org/2002/ws/addr/>

¹⁵ <http://www.w3.org/TR/wsdl>

¹⁹ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf

²⁰ <http://www.w3.org/XML/>