

GT 4.2.1 Component Guide to Public Interfaces: C WS Core

GT 4.2.1 Component Guide to Public Interfaces: C WS Core

Table of Contents

1. APIs	1
1. Programming Model Overview	1
2. Component API	1
I. C WS Core Commands	3
globus-wsc-container	4
globus-wsrf-cgen	6
globus-wsrf-destroy	10
globus-wsrf-set-termination-time	12
globus-wsrf-query	14
globus-wsrf-get-property	17
globus-wsrf-get-properties	19
globus-wsrf-insert-property	21
globus-wsrf-update-property	24
globus-wsrf-delete-property	27
globus-wsn-get-current-message	29
globus-wsn-pause-subscription	32
globus-wsn-resume-subscription	34
globus-wsn-subscribe	36
A. Errors	39
Glossary	41

List of Tables

1. WSRF Core Namespaces and C Prefixes	8
2. Common options	10
3. Common options	12
4. Application-specific options	14
5. Common options	15
6. Common options	17
7. Common options	19
8. Common options	22
9. Common options	25
10. Common options	27
11. Application-specific options	29
12. Common options	30
13. Common options	32
14. Common options	34
15. Application-specific options	36
16. Common options	37
A.1. C WS Core Errors	40

Chapter 1. APIs

1. Programming Model Overview

The C WS-Core provides interfaces for developers interested in writing web services and clients in C. The primary APIs available to the developer are the C stub bindings generated from WSDL and XSD. These APIs provide the structures and type definitions for each XML Schema type, client stub functions for invoking services, and service skeleton code that allows service writers to fill in the service implementation.

The client stub bindings provide the following:

- Portable ANSI-C API
- Control of message handling and configurable attributes through client handles
- Asynchronous stub functions for non-blocking requests
- EPR encapsulation for easy interaction with resources
- Convenient handling of XSD wildcards

For service writers, the C WS-Core provides service-side skeleton bindings that perform the necessary routing and marshalling for a service operation. The interface to the developer is through the service implementation functions that must be filled in. The service-side programming model includes the ability to load operation providers, which are generic operation implementations that exist over a set of services. This is useful with WSRF, where pseudo-operation inheritance exists. As well, message handling can be controlled at the service implementation level, providing flexibility and control to the service developer.

The C WS-Core provides resource management using the resource API. This is a C API that can be invoked from within C services for creation, access, and control of resources and resource properties.

2. Component API

- [Resource and Resource Property API](http://www.globus.org/api/c-globus-4.2.1/globus_c_wsrf_resource/html/index.html)¹: Useful for writing WSRF-enabled services. This API allows resources to be created, accessed, and modified from within a C Web Service implementation.
- [Service Engine and Message Attributes](http://www.globus.org/api/c-globus-4.2.1/globus_c_ws_messaging/html/index.html)²: The message attributes provides mechanisms for manipulating runtime parameters of messages. This includes security setup, specific HTTP and WS-Addressing configuration, among others.

The service engine API is useful for embedding Web Services in C programs. This API allows an application to directly control service invocations and interact with services as they are being invoked. It also provides a convenient API for running a NotificationConsumer service (receiving notifications) from within a client application.

- [Notification Consumer API](http://www.globus.org/api/c-globus-4.2.1/globus_notification_consumer/html/index.html)³: Allows creation of NotificationConsumer resource instances from a client API. This API can be used in combination with the Service Engine API to receive notifications.

¹ http://www.globus.org/api/c-globus-4.2.1/globus_c_wsrf_resource/html/index.html

² http://www.globus.org/api/c-globus-4.2.1/globus_c_ws_messaging/html/index.html

³ http://www.globus.org/api/c-globus-4.2.1/globus_notification_consumer/html/index.html

- WSRF Core Bindings API⁴: These are the types generated from the set of core WSRF schemas. For example, the *wsa_EndpointReferenceType* passed to all EPR stub functions is a generated type from the WS-Addressing schema. The other schemas include:
 - WS-Addressing
 - WS-BaseFaults
 - WS-ResourceProperties
 - WS-ResourceLifetime
 - WS-BaseN
 - WS-ServiceGroup

⁴ http://www.globus.org/api/c-globus-4.2.1/globus_c_wsrp_core_bindings/html/index.html

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-FILE</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 1. 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/xmldsig#	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 2. 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 3. 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 4. 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 5. 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 6. 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 7. 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 8. 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 9. 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 10. 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 11. 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 12. 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 13. 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 14. 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:A  
  <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-BaseNotif  
      ns01:type="ns02:string">7d6430e4-f019-11da-1b9-00096b86f788</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 15. 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 16. 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.

Appendix A. Errors

Table A.1. C WS Core Errors

Error Code	Definition	Possible Solutions
globus_soap_message_module: Failed sending request http://widgets.com/WidgetPortType/createWidgetRequest. globus_xio: Unable to connect to grid.example.org:8080 globus_xio: System error in connect: Connection refused globus_xio: A system call failed: Connection refused	Unable to contact service container	Check that the service endpoint refers to a running container.
globus_soap_message_module: Failed sending request http://widgets.com/WidgetPortType/createWidgetRequest. globus_xio_gsi: gss_init_sec_context failed. GSS Major Status: Unexpected Gatekeeper or Service Name globus_gsi_gss_api: Authorization denied: The name of the remote entity (/C=US/O=Globus Alliance/OU=Service/CN=host/grid.example.org), and the expected name for the remote entity (/C=US/O=Globus Alliance/OU=Service/CN=host/cloud.example.org) do not match	Service is not running with the expected security credential.	Verify that the service credential being presented by the service (first parenthesized name) is a reasonable certificate name for the service. If so, set the GLOBUS_SOAP_MESSAGE_PEER_IDENTITY_KEY attribute on the soap message handle to that identity. For most command-line wsrfl tools, this can be done by passing it as an argument to the -z command-line parameter.
globus_soap_message_module: SOAP Fault Fault code: Client Fault string: globus_service_engine_module: Failed to find operation: {XXXX}YYYY for service: {ZZZZ}BBBB	The service port type {ZZZZ}BBBB does not contain a {XXXX}YYYY operation.	Verify that the client bindings are built from the same WSDL and XML Schema documents as the service.
globus_soap_message_module: Failed receiving response http://widgets.com/WidgetPortType/createWidgetResponse. ws_addressing: Addressing header is a draft version of WS Addressing: "http://schemas.xmlsoap.org/ws/2004/03/addressing". This could be a GT version mismatch, client is GT 4.2.x and response is from GT 4.0.x server	The service is running on a container which is using a draft version of the WS-Addressing specification. This was used by GT 4.0.x	Update the service to work with GT 4.2.x or compile your client with GT 4.0.x libraries.
globus_soap_message_module: Failed sending request http://widgets.com/WidgetPortType/createWidgetRequest. globus_xio: The GSI XIO driver failed to establish a secure connection. The failure occurred during a handshake read. globus_xio: An end of file occurred	The service container either did not support SSL authentication, or the service container did not trust the client certificate	Consult the service administrator to verify that the service container supports SSL and that your certificate is issued by a certificate authority trusted by the service.

Glossary

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.

S

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.

T

transport-level security Uses transport-level security (TLS) mechanisms.

W

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.

¹³ <http://www.w3.org/TR/soap/>

¹⁵ <http://www.w3.org/TR/wsdl>

¹⁹ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf