GT 6.0 C Common Libraries: Developer's Guide

GT 6.0 C Common Libraries: Developer's Guide

Introduction

The C Common Libraries provide an abstraction layer for data types, libc system calls, and data structures used throughout the Globus Toolkit and useful for applications that use the Globus Toolkit.

Table of Contents

List of Tables

Chapter 1. Before you begin

1. Feature summary

- Globus Callback - Portable event handling layer for signal handling and periodic and one-shot events in a single- more multi-threaded environment.
- Globus Error - An abstraction for providing context-specific information in error response in C.
- Portable Threading API for POSIX and Windows
- URL String Parser
- Configuration handlers for command-line, environment-variable, and configuration file based application configuration.

2. Tested platforms

The C common libraries have been tested on the following platforms

Table 1.1. Tested Platforms

Operating System	Distribution	Version(s)	Architecture(s)
	CentOS	5	i386, x86_64
	Fedora	18, 19	i386, x86_64
	Red Hat Enterprise Linux	5, 6	i386, x86_64
	Scientific Linux	5, 6	i386, x86_64
	Debian	6, 7	i386, amd64
	Ubuntu	10.04LTS, 12.04LTS, 12.10, 13.04, 13.10	i386, amd64
Mac OS X		10.8 (Mountain Lion)	x86_64
Solaris		11	x86_64

3. Backward compatibility summary

API changes since GT version 5.2.5

None.

All of the GT 3.2 API is still functional in GT 6.0.

4. Technology dependencies

C Common Libraries only depend on the globus_core module.

5. Security Considerations for C Common Libraries

There are no security considerations for the C Common Libraries.

Chapter 2. Usage scenarios

C Common libraries will need to be used if virtually any other toolkit component is used, since many data types are abstract and require the C common libraries to manipulate.

Chapter 3. Architecture and design overview

Not available at this time.

Chapter 4. APIs

1. Component API

See the C API pages[1] for other API documentation on globus_common.

[1] http://www.globus.org/api/c-globus-6.0/

Chapter 5. Environment variable interface

1. Environment variables for C Common Libraries

GLOBUS_HOSTNAME	Set this variable to the fully qualified name of the local machine's hostname.
GLOBUS_DOMAIN_NAME	Set this variable to the domain name to be used to qualify the local machine's hostname.
GLOBUS_ERROR_OUTPUT	Set this variable to 1 to cause Globus libraries to display error information to stderr.
GLOBUS_ERROR_VERBOSE	Set this variable to 1 to enable verbose error messages.
GLOBUS_I18N	Set this variable to 1 to attempt to use localized messages. (Currently not working)
GLOBUS_LOCATION	Set this variable to the path where the Globus Toolkit is installed, so that Globus tools can find libraries and data files. This is only needed if the Globus Toolkit was built with the source installer.
GLOBUS_THREAD_MODEL	Set to the name of a thread model to control the operation of the Globus event driver. Valid values are (depending on the platform) `none` for non-threaded operation (the default), `pthread` for POSIX threads, or `windows` for Windows threads.

Chapter 6. Debugging

General C debugging techniques apply when developing with the C common libraries.

Chapter 7. Troubleshooting

There are no specific troubleshooting techniques for the C common libraries.

Chapter 8. Related Documentation

See the C API pages[1] for more information about this component.

[1] http://www.globus.org/api/c-globus-6.0/

GT 6.0 Migrating Guide for C Common Libraries

Table of Contents

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

1. Migrating C Common Libraries from GT5.0

All components are compiled with a runtime configuration of thread model, instead of having to compile both threaded and nonthreaded build flavors. To enable threads for a program, set the environment variable `GLOBUS_THREAD_MODEL` or call `globus_thread_set_model`() with the desired thread model, such as `pthread` or `windows`.

2. Migrating C Common Libraries from GT4.2

No changes need to be made in applications using version 4.2.x releases of the C common libraries.

3. Migrating C Common Libraries from GT4.0

No changes need to be made in applications using version 4.0.x releases of the C common libraries.

4. Migrating C Common Libraries from GT3

No changes need to be made in applications using version 3.x releases of the C common libraries.

5. Migrating C Common Libraries from GT2

No changes need to be made in applications using version 2.x releases of the C common libraries.

GT 6.0 C Common Libraries: Quality Profile

Table of Contents

1. Test coverage reports

There are no reports on this component.

2. Code analysis reports

There are no reports on this component.

3. Known Problems in C Common Libraries

- GT-108[1]: --libdir is being ignored
- GT-114[2]: i18n rules in installer don't work
- GT-360[3]: signals ignored by subprocesses
- GT-452[4]: autoconf macros for globus-core can't handle absolute path in CC variable (e.g. CC=/usr/bin/gcc)
- GT-470[5]: Globus IO reports timeout error as cancellation

4. Fixed Bugs for C Common Libraries

- GT-390[6]: globus_callout loader fails load plugins on systems with broken libtool-ltdl
- GT-470[7]: Globus IO reports timeout error as cancellation

5. Performance reports

There are no reports on this component.

GT 6.0 Release Notes: C Common Libraries

Table of Contents

1. Component Overview

The C Common Libraries provide an abstraction layer for data types, libc system calls, and data structures used throughout the Globus Toolkit and useful for applications that use the Globus Toolkit.

2. Feature summary

- Globus Callback - Portable event handling layer for signal handling and periodic and one-shot events in a single- more multi-threaded environment.
- Globus Error - An abstraction for providing context-specific information in error response in C.
- Portable Threading API for POSIX and Windows
- URL String Parser
- Configuration handlers for command-line, environment-variable, and configuration file based application configuration.

3. Summary of Changes in C Common Libraries

3.1. New Features: C Common Libraries

None.

3.2. Improvements: C Common Libraries

- Update of build system to enable parallel builds, cross-compiling, and easier testing

4. Fixed Bugs for C Common Libraries

- GT-390[1]: globus_callout loader fails load plugins on systems with broken libtool-ltdl
- GT-470[2]: Globus IO reports timeout error as cancellation

5. Known Problems in C Common Libraries

- GT-108[3]: --libdir is being ignored
- GT-114[4]: i18n rules in installer don't work
- GT-360[5]: signals ignored by subprocesses
- GT-452[6]: autoconf macros for globus-core can't handle absolute path in CC variable (e.g. CC=/usr/bin/gcc)
- GT-470[7]: Globus IO reports timeout error as cancellation

6. Technology dependencies

C Common Libraries only depend on the globus_core module.

7. Tested platforms

The C common libraries have been tested on the following platforms

Table 1. Tested Platforms

Operating System	Distribution	Version(s)	Architecture(s)
	CentOS	5	i386, x86_64
	Fedora	18, 19	i386, x86_64
	Red Hat Enterprise Linux	5, 6	i386, x86_64
	Scientific Linux	5, 6	i386, x86_64
	Debian	6, 7	i386, amd64
	Ubuntu	10.04LTS, 12.04LTS, 12.10, 13.04, 13.10	i386, amd64
Mac OS X		10.8 (Mountain Lion)	x86_64
Solaris		11	x86_64

8. Backward compatibility summary

API changes since GT version 5.2.5

None.

All of the GT 3.2 API is still functional in GT 6.0.

9. Associated Standards

There are no standards implemented by the C common libraries.

10. For More Information

See the C API pages[1] for more information about this component.

[1] http://www.globus.org/api/c-globus-6.0/