

## **GT 4.0: C Common Libraries**

---

## **GT 4.0: C Common Libraries**

---

---

# Table of Contents

|                                         |    |
|-----------------------------------------|----|
| 1. Key Concepts .....                   | 1  |
| 1. Overview .....                       | 1  |
| 2. Conceptual Details .....             | 1  |
| 3. Related Documents .....              | 6  |
| 2. 4.0.0 Release Notes .....            | 7  |
| 1. Component Overview .....             | 7  |
| 2. Feature Summary .....                | 7  |
| 3. Bug Fixes .....                      | 7  |
| 4. Known Problems .....                 | 7  |
| 5. Technology Dependencies .....        | 7  |
| 6. Tested Platforms .....               | 7  |
| 7. Backward Compatibility Summary ..... | 7  |
| 8. For More Information .....           | 8  |
| 3. 4.0.1 Release Notes .....            | 9  |
| 1. Introduction .....                   | 9  |
| 2. Changes Summary .....                | 9  |
| 3. Bug Fixes .....                      | 9  |
| 4. Known Problems .....                 | 9  |
| 5. For More Information .....           | 10 |
| 4. 4.0.2 Release Notes .....            | 11 |
| 1. Introduction .....                   | 11 |
| 2. Changes Summary .....                | 11 |
| 3. Bug Fixes .....                      | 11 |
| 4. Known Problems .....                 | 11 |
| 5. For More Information .....           | 13 |
| 5. 4.0.3 Release Notes .....            | 14 |
| 1. Introduction .....                   | 14 |
| 2. Changes Summary .....                | 14 |
| 3. Bug Fixes .....                      | 14 |
| 4. Known Problems .....                 | 14 |
| 5. For More Information .....           | 15 |
| 6. 4.0.4 Release Notes .....            | 16 |
| 1. Introduction .....                   | 16 |
| 2. Changes Summary .....                | 16 |
| 3. Bug Fixes .....                      | 16 |
| 4. Known Problems .....                 | 16 |
| 5. For More Information .....           | 17 |
| 7. System Administrator's Guide .....   | 18 |
| 1. Introduction .....                   | 18 |
| 2. Building and Installing .....        | 18 |
| 3. Configuring .....                    | 18 |
| 4. Deploying .....                      | 18 |
| 5. Testing .....                        | 18 |
| 6. Security Considerations .....        | 18 |
| 7. Troubleshooting .....                | 18 |
| 8. User's Guide .....                   | 19 |
| 1. Introduction .....                   | 19 |
| 2. Command-line tools .....             | 19 |
| 3. Troubleshooting .....                | 19 |
| 9. Developer's Guide .....              | 20 |
| 1. Introduction .....                   | 20 |

|                                                            |    |
|------------------------------------------------------------|----|
| 2. Before you begin .....                                  | 20 |
| 3. Architecture and design overview .....                  | 20 |
| 4. Public interface .....                                  | 21 |
| 5. Usage scenarios .....                                   | 21 |
| 6. Debugging .....                                         | 21 |
| 7. Troubleshooting .....                                   | 21 |
| 8. Related Documentation .....                             | 21 |
| 10. Fact Sheet .....                                       | 22 |
| 1. Brief component overview .....                          | 22 |
| 2. Summary of features .....                               | 22 |
| 3. Usability summary .....                                 | 22 |
| 4. Backward compatibility summary .....                    | 22 |
| 5. Technology dependencies .....                           | 22 |
| 6. Tested platforms .....                                  | 22 |
| 7. Associated standards .....                              | 22 |
| 8. For More Information .....                              | 23 |
| 11. Public Interface Guide .....                           | 24 |
| 1. Semantics and syntax of APIs .....                      | 24 |
| 2. Semantics and syntax of the WSDL .....                  | 25 |
| 3. Command-line tools .....                                | 25 |
| 4. Overview of Graphical User Interface .....              | 25 |
| 5. Semantics and syntax of domain-specific interface ..... | 25 |
| 6. Configuration interface .....                           | 25 |
| 7. Environment variable interface .....                    | 25 |
| 12. Quality Profile .....                                  | 26 |
| 1. Test coverage reports .....                             | 26 |
| 2. Code analysis reports .....                             | 26 |
| 3. Outstanding bugs .....                                  | 26 |
| 4. Bug Fixes .....                                         | 26 |
| 5. Performance reports .....                               | 26 |
| 13. Migrating Guide .....                                  | 27 |
| 1. Migrating from GT2 .....                                | 27 |
| 2. Migrating from GT3 .....                                | 27 |

---

# Chapter 1. GT 4.0 Common Runtime Components: Key Concepts

## 1. Overview

The common runtime components provide GT4 web and pre-web services with a set of libraries and tools that allows these services to be platform independent, to build on various abstraction layers (threading, io) and to leverage functionality lower in the web services stack (WSRF, WSN, etc).

These components are architecturally diverse and it is thus hard to identify a overarching theme. Instead a few sub-themes have been identified and elaborated on in the below.

## 2. Conceptual Details

### 2.1. Web Services

We introduce basic concepts relating to Web services and their use and implementation within GT4, in particular within the "WS Core" (Java & C) components.

#### 2.1.1. GT4, Distributed Systems, and Web Services

GT4 is a set of software components for building distributed systems: systems in which diverse and discrete software agents interact via message exchanges over a network to perform some tasks. Distributed systems face particular challenges relating to sometimes high and unpredictable network latencies, the possibility of partial failure, and issues of concurrency. In addition, system components may be located within distinct administrative domains, thus introducing issues of decentralized control and negotiation.

GT4 is, more specifically, a set of software components that (with some exceptions) implement Web services mechanisms for building distributed systems. Web services provide a standard means of interoperating between different software applications running on a variety of platforms and/or frameworks.

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Web services standardize the messages that entities in a distributed system must exchange in order to perform various operations. At the lowest level, this standardization concerns the protocol used to transport messages (typically HTTP), message encoding (SOAP), and interface description (WSDL). A client interacts with a Web service by sending it a SOAP message; the client may subsequently receive response message(s) in reply. At higher levels, other specifications define conventions for securing message exchanges (e.g., WS-Security), for management (e.g., WSDM), and for higher-level functions such as discovery and choreography. Figure 1 presents a view of these different component technologies; we discuss specific specifications below in [Section 2.1.4, "Web Services Specifications"](#).

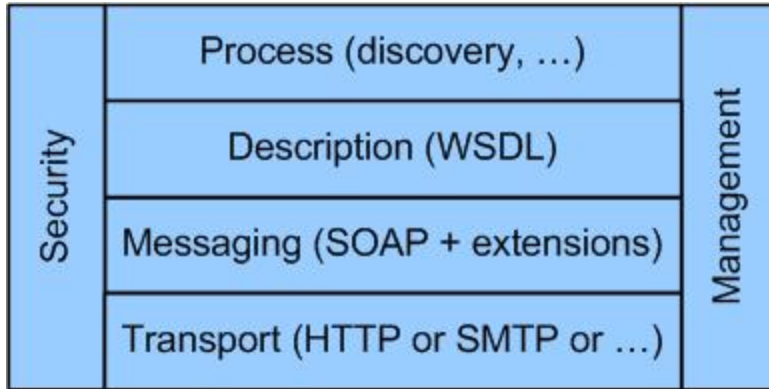


Figure 1: An abstract view of the various specifications that define the Web services architecture

### 2.1.2. Service Oriented Applications and Infrastructure

Web services technologies, and GT4 in particular, can be used to build both service-oriented applications and service-oriented infrastructure. Deferring discussion of the sometimes controversial term "service-oriented" to later in [Section 2.1.9, "Service Oriented Architecture"](#), we note that a service-oriented application is constructed via the composition of components defined by service interfaces (in the current context, Web services): for example, a financial or biological database, an options pricing routine, or a biological sequence analyzer. Many descriptions of Web services and SOAP focus on the task of defining interfaces to such components, often illustrating their discussion with examples such as a "stock quote service" (the "hello world" of Web services).

Particularly when servicing many such requests from a distributed community, we face the related problem of orchestrating and managing numerous distributed hardware and software components. Web services can be used for this purpose also, and thus we introduce the term service-oriented infrastructure to denote the resource management and provisioning mechanisms used to meet quality of service goals for components and applications. Many GT4 features are concerned with enabling the construction of service-oriented infrastructure.

### 2.1.3. Web Services Implementation

From the client perspective, a Web service is simply a network-accessible entity that processes SOAP messages. Things are somewhat more complex under the covers. To simplify service implementation, it is common for a Web services implementation to distinguish between:

1. the hosting environment (or container), the (domain-independent) logic used to receive a SOAP message and identify and invoke the appropriate code to handle the message, and potentially also to provide related administration functions, and;
2. the Web service implementation, the (domain-specific) code that handles the message.

This separation of concerns means that the developer need only provide the domain-specific message handling code to implement a new service. It is also common to further partition the hosting environment logic into that concerned with transporting the SOAP message (typically via HTTP, thus an "HTTP engine" or "Web server"-sometimes termed an "application server") and that concerned with processing SOAP messages (the "SOAP engine" or "SOAP processor"). Figure 2 illustrates these various components.

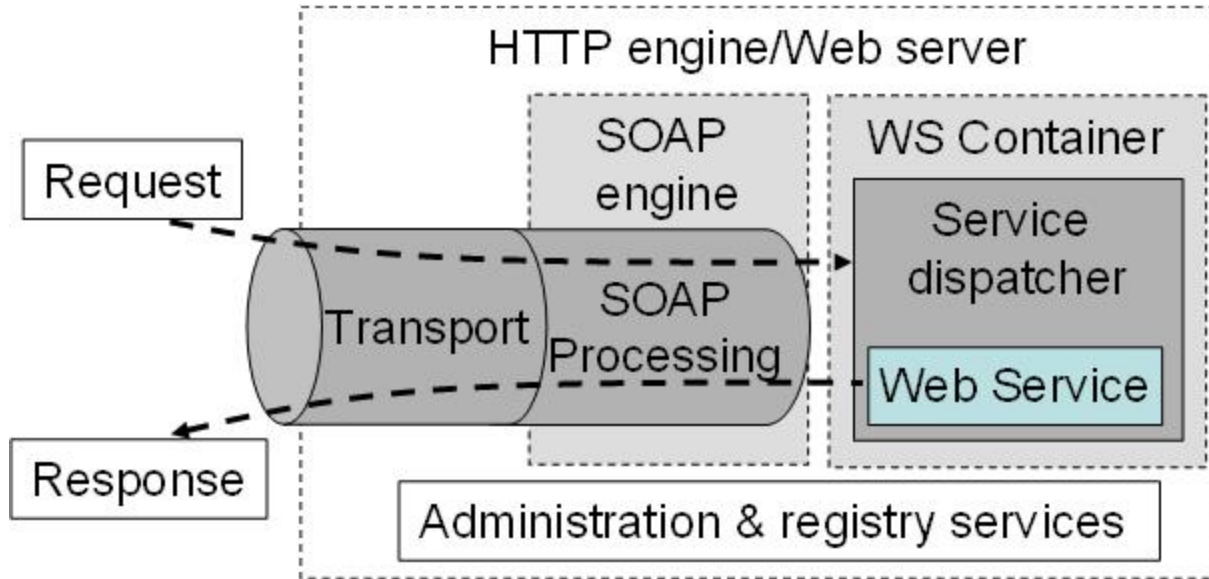


Figure 2: WS Container. High-level picture of functional components commonly encountered in Web service implementations, showing the path taken by requests and responses.

Many different containers exist, with different performance properties, supported Web services implementation languages, security support, and so forth. We mention below those used in GT4.

### 2.1.4. Web Services Specifications

We provide pointers to the Web services specifications that underlie GT4. These comprise the core specifications that define the Web services architecture (XML, SOAP, WSDL); WS-Security and other specifications relating to security; and the WS-Addressing, WSRF, and WS-Notification specifications used to define, name, and interact with stateful resources. We also speak briefly to emerging specifications that are likely to be important in future GT evolution. An important source of information on the requirements that motivate the use and development of these specifications is the Open Grid Services Architecture.

### 2.1.5. XML, SOAP, WSDL

XML is used extensively within Web services as a standard, flexible, and extensible data format. In addition to XML syntax, other important specifications are XML Schema and XML Namespaces. Note that while current Web services tools typically adopt a textual serialization, a binary encoding is also possible and may provide higher efficiency.

SOAP 1.2 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.

WSDL 1.1 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.

### 2.1.6. WS-Security and Friends

The WS-Security family of specifications addresses a range of issues relating to authentication, authorization, policy representation, and trust negotiation in a Web services context. GT4 uses a number of these specifications plus other related specifications, notably Security Authorization Markup Language (SAML), to address message protection, authentication, delegation, and authorization, as follows:

- TLS (transport-level) or WS-Security and WS-SecureConversation (message level) are used as message protection mechanisms in combination with SOAP.
- X.509 End Entity Certificates or Username and Password are used as authentication credentials.
- X.509 Proxy Certificates and WS-Trust are used for delegation.
- SAML assertions are used for authorization.

### 2.1.7. WS-Addressing, WSRF, and WS-Notification

A number of related specifications provide functionality important for service oriented infrastructure in which we need to be able to represent and manipulate stateful entities such as physical resources of various kinds, logical components such as software licenses, and transient activities such as tasks and workflows.

The WS-Addressing specification defines transport-neutral mechanisms to address Web services and messages. Specifically, this specification defines XML elements to identify Web service endpoints and to secure end-to-end endpoint identification in messages.

The WS Resource Framework (WSRF) specifications define a generic and open framework for modeling and accessing stateful resources using Web services. 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).

The WS-Notification family of specifications define a pattern-based approach to allowing Web services to disseminate information to one another. This framework comprises mechanisms for basic notification (WS-Notification), topic-based notification (WS-Topics), and brokered notification (WS-BrokeredNotification).

We note that the Web services standards space is in some turmoil due to competing proposed specifications. In particular, Microsoft and others recently proposed WS-Transfer, WS-Eventing, and WS-Management, which define similar functionality to WSRF, WS-Notification, and WSDM (discussed below), respectively, but using different syntax. We hope that these differences will be resolved in the future.

### 2.1.8. Other Relevant Specifications

The WS-Interoperability (WS-I) organization has produced a number of profiles that define ways in which existing Web services specifications can be used to promote interoperability among different implementations. The WS-I Basic Profile speaks to messaging and service description: primarily XML, SOAP, and WSDL. The WS-I Basic Security Profile speaks to basic security mechanisms. Other profiles are under development.

Web services distributed management (WSDM) specifications under development within OASIS are likely to play a role in future GT implementations as a means of managing GT components.

WS-CIM specifications under development within DMTF are likely to play a role in future GT implementations as a means of representing physical and virtual resources.

The Global Grid Forum's Open Grid Services Architecture (OGSA) working group has completed a document that provides a high-level description of the functionality required for future service-oriented infrastructure and applications, and a framework that suggests how this functionality can be factored into distinct specifications. The OGSA working group is now proceeding to define OGSA Profiles that, like WS-I profiles, will identify technical specifications that can be used to address specific Grid scenarios.

### 2.1.9. Service Oriented Architecture

We provide some additional discussion concerning the term service oriented architecture (SOA), which is used widely but not necessarily consistently within the Web services community. One common usage is simply to indicate the use of Web services technologies. However, the intention of those who coined the term seems to be rather to contrast two different styles of building distributed systems. Distributed object systems are distributed systems in which the semantics of object initialization and method invocation are exposed to remote systems by means of a proprietary or standardized mechanism to broker requests across system boundaries, marshal and unmarshal method argument data, etc. Distributed objects systems typically (albeit not necessarily) are characterized by objects maintaining a fairly complex internal state required to support their methods, a fine grained or "chatty" interaction between an object and a program using it, and a focus on a shared implementation type system and interface hierarchy between the object and the program that uses it.

In contrast, a Service Oriented Architecture (SOA) is a form of distributed systems architecture that is typically characterized by the following properties:

- Logical view: The service is an abstracted, logical view of actual programs, databases, business processes, etc., defined in terms of what it does, typically carrying out a business-level operation.
- Message orientation: The service is formally defined in terms of the messages exchanged between provider agents and requester agents, and not the properties of the agents themselves. The internal structure of an agent, including features such as its implementation language, process structure and even database structure, are deliberately abstracted away in the SOA: using the SOA discipline one does not and should not need to know how an agent implementing a service is constructed. A key benefit of this concerns so-called legacy systems. By avoiding any knowledge of the internal structure of an agent, one can incorporate any software component or application that can be "wrapped" in message handling code that allows it to adhere to the formal service definition.
- Description orientation: A service is described by machine-processable metadata. The description supports the public nature of the SOA: only those details that are exposed to the public and important for the use of the service should be included in the description. The semantics of a service should be documented, either directly or indirectly, by its description.
- Granularity: Services tend to use a small number of operations with relatively large and complex messages.
- Network orientation: Services tend to be oriented toward use over a network, though this is not an absolute requirement.
- Platform neutral: Messages are sent in a platform-neutral, standardized format delivered through the interfaces. XML is the most obvious format that meets this constraint.

It is argued that these features can allow service-oriented architectures to cope more effectively with issues that arise in distributed systems, such as problems introduced by latency and unreliability of the underlying transport, the lack of shared memory between the caller and object, problems introduced by partial failure scenarios, the challenges of concurrent access to remote resources, and the fragility of distributed systems if incompatible updates are introduced to any participant.

Web services technologies in general, and GT4 in particular, can be used to build both distributed object systems and service-oriented architectures. The specific design principles to be followed in a particular setting will depend on a variety of issues, including target environment, scale, platform heterogeneity, and expected future evolution.

## 3. Related Documents

### 3.1. Web Services

- Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C. and Orchard, D. Web Services Architecture. W3C, Working Draft<sup>1</sup>

---

<sup>1</sup> <http://www.w3.org/TR/2003/WD-ws-arch-20030808/>

---

# Chapter 2. GT 4.0 Release Notes: C Common Libraries

## 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

Features new in release GT 4.0

- globus\_range\_list abstraction added
- globus\_logging abstraction added

## 3. Bug Fixes

See a [bugzilla](http://bugzilla.globus.org/globus/query.cgi)<sup>1</sup> query on Toolkit Internals->globus\_common to list the bugs fixed.

## 4. Known Problems

See a [bugzilla](http://bugzilla.globus.org/globus/query.cgi)<sup>2</sup> query on Toolkit Internals->globus\_common to list the bugs outstanding.

## 5. Technology Dependencies

C Common Libraries only depend on the globus\_core module.

## 6. Tested Platforms

The C common libraries work on any platform on which the toolkit is supported. See [supported platforms](http://www.globus.org/toolkit/docs/4.0/admin/docbook/ch03.html#s-platform)<sup>3</sup>.

## 7. Backward Compatibility Summary

API changes since GT version 3.2

- globus\_range\_list abstraction added
- globus\_logging abstraction added

All of the GT 3.2 API is still functional in GT 4.0

---

<sup>1</sup> <http://bugzilla.globus.org/globus/query.cgi>

<sup>2</sup> <http://bugzilla.globus.org/globus/query.cgi>

<sup>3</sup> <http://www.globus.org/toolkit/docs/4.0/admin/docbook/ch03.html#s-platform>

## 8. For More Information

See the [C API pages](http://www.globus.org/api/c-globus-4.0/)<sup>4</sup> for more information about this component.

---

<sup>4</sup> <http://www.globus.org/api/c-globus-4.0/>

---

# Chapter 3. GT 4.0.1 Incremental Release Notes: C Common Libraries

## 1. Introduction

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

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [C Common Libraries 4.0 Release Notes](#)<sup>1</sup>.

## 2. Changes Summary

No changes have occurred for C Common Libraries.

## 3. Bug Fixes

No bugs were fixed for C Common Libraries.

## 4. Known Problems

The following problems are known to exist for C Common Libraries at the time of the 4.0.1 release:

- [Bug 484](#):<sup>2</sup> Mem leak on deactivate\_all
- [Bug 686](#):<sup>3</sup> globus\_mutex\_lock() behaves differently on Unix -vs- Win32
- [Bug 930](#):<sup>4</sup> inttypes.h included without testing if it's present
- [Bug 1141](#):<sup>5</sup> Misleading error messages using globus-makefile-header
- [Bug 2315](#):<sup>6</sup> Globus common fails to compile if there is no ipv6 support
- [Bug 2791](#):<sup>7</sup> Command globus-domainname needs g11n
- [Bug 2793](#):<sup>8</sup> Command globus-generalized-time needs g11n
- [Bug 2794](#):<sup>9</sup> Command globus-hostname needs g11n

---

<sup>1</sup> [http://www.globus.org/toolkit/docs/4.0/common/ccommonlib/C\\_Common\\_Libraries\\_Release\\_Notes.html](http://www.globus.org/toolkit/docs/4.0/common/ccommonlib/C_Common_Libraries_Release_Notes.html)

<sup>2</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=484](http://bugzilla.globus.org/globus/show_bug.cgi?id=484)

<sup>3</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=686](http://bugzilla.globus.org/globus/show_bug.cgi?id=686)

<sup>4</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=930](http://bugzilla.globus.org/globus/show_bug.cgi?id=930)

<sup>5</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=1141](http://bugzilla.globus.org/globus/show_bug.cgi?id=1141)

<sup>6</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2315](http://bugzilla.globus.org/globus/show_bug.cgi?id=2315)

<sup>7</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2791](http://bugzilla.globus.org/globus/show_bug.cgi?id=2791)

<sup>8</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2793](http://bugzilla.globus.org/globus/show_bug.cgi?id=2793)

<sup>9</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2794](http://bugzilla.globus.org/globus/show_bug.cgi?id=2794)

- [Bug 2801](#):<sup>10</sup> Command globus-version needs g11n
- [Bug 3081](#):<sup>11</sup> Small leak in libltdl
- [Bug 3107](#):<sup>12</sup> Fatal error: tcp\_init(): globus\_io\_tcp\_create\_listener() ...
- [Bug 3350](#):<sup>13</sup> globusrun-ws hangs for 2 minutes at startup
- [Bug 3380](#):<sup>14</sup> doxygen-related scripts belong in core?
- [Bug 3392](#):<sup>15</sup> globusrun-ws Bus error
- [Bug 3427](#):<sup>16</sup> Issues with permission of file testcred.pem located in gl...
- [Bug 3436](#):<sup>17</sup> globus\_uuid MAC address retrieval doesn't work on solaris
- [Bug 3509](#):<sup>18</sup> GLOBUS\_COMMON\_MODULE->GLOBUS\_CALLBACK\_MODULE deactivation...
- [Bug 3610](#):<sup>19</sup> globus-makefile-header returns bad field

## 5. For More Information

Click [here](#)<sup>20</sup> for more information about this component.

---

<sup>10</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2801](http://bugzilla.globus.org/globus/show_bug.cgi?id=2801)

<sup>11</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3081](http://bugzilla.globus.org/globus/show_bug.cgi?id=3081)

<sup>12</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3107](http://bugzilla.globus.org/globus/show_bug.cgi?id=3107)

<sup>13</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3350](http://bugzilla.globus.org/globus/show_bug.cgi?id=3350)

<sup>14</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3380](http://bugzilla.globus.org/globus/show_bug.cgi?id=3380)

<sup>15</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3392](http://bugzilla.globus.org/globus/show_bug.cgi?id=3392)

<sup>16</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3427](http://bugzilla.globus.org/globus/show_bug.cgi?id=3427)

<sup>17</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3436](http://bugzilla.globus.org/globus/show_bug.cgi?id=3436)

<sup>18</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3509](http://bugzilla.globus.org/globus/show_bug.cgi?id=3509)

<sup>19</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3610](http://bugzilla.globus.org/globus/show_bug.cgi?id=3610)

<sup>20</sup> [index.html](#)

---

# Chapter 4. GT 4.0.2 Incremental Release Notes: C Common Libraries

## 1. Introduction

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

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [C Common Libraries 4.0 Release Notes](#)<sup>1</sup>.

## 2. Changes Summary

No changes have occurred for C Common Libraries.

## 3. Bug Fixes

The following bugs were fixed in the C Common Libraries:

- [Bug 3392](#):<sup>2</sup> globusrun-ws Bus error
- [Bug 3427](#):<sup>3</sup> Issues with permission of file testcred.pem located in gl...
- [Bug 3624](#):<sup>4</sup> Linux on sparc64 hw support
- [Bug 3713](#):<sup>5</sup> i18n does not build on HP-UX/IA-64 platform
- [Bug 3936](#):<sup>6</sup> Nightly build failure on AIX in trunk
- [Bug 3946](#):<sup>7</sup> don't set \$SASL\_PATH unnecessarily
- [Bug 4240](#):<sup>8</sup> compilation error when building globus\_debug.c against MP...
- [Bug 4309](#):<sup>9</sup> HP-UX IA64 broken due to not hardcoding libpaths

## 4. Known Problems

The following problems are known to exist for C Common Libraries at the time of the 4.0.2 release:

---

<sup>1</sup> [http://www.globus.org/toolkit/docs/4.0/common/ccommonlib/C\\_Common\\_Libraries\\_Release\\_Notes.html](http://www.globus.org/toolkit/docs/4.0/common/ccommonlib/C_Common_Libraries_Release_Notes.html)

<sup>2</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3392](http://bugzilla.globus.org/globus/show_bug.cgi?id=3392)

<sup>3</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3427](http://bugzilla.globus.org/globus/show_bug.cgi?id=3427)

<sup>4</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3624](http://bugzilla.globus.org/globus/show_bug.cgi?id=3624)

<sup>5</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3713](http://bugzilla.globus.org/globus/show_bug.cgi?id=3713)

<sup>6</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3936](http://bugzilla.globus.org/globus/show_bug.cgi?id=3936)

<sup>7</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3946](http://bugzilla.globus.org/globus/show_bug.cgi?id=3946)

<sup>8</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4240](http://bugzilla.globus.org/globus/show_bug.cgi?id=4240)

<sup>9</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4309](http://bugzilla.globus.org/globus/show_bug.cgi?id=4309)

- [Bug 484](#):<sup>10</sup> Mem leak on deactivate\_all
- [Bug 686](#):<sup>11</sup> globus\_mutex\_lock() behaves differently on Unix -vs- Win32
- [Bug 930](#):<sup>12</sup> inttypes.h included without testing if it's present
- [Bug 1141](#):<sup>13</sup> Misleading error messages using globus-makefile-header
- [Bug 2315](#):<sup>14</sup> Globus common fails to compile if there is no ipv6 support
- [Bug 2791](#):<sup>15</sup> Command globus-domainname needs g11n
- [Bug 2793](#):<sup>16</sup> Command globus-generalized-time needs g11n
- [Bug 2794](#):<sup>17</sup> Command globus-hostname needs g11n
- [Bug 2801](#):<sup>18</sup> Command globus-version needs g11n
- [Bug 3081](#):<sup>19</sup> Small leak in libltdl
- [Bug 3107](#):<sup>20</sup> Fatal error: tcp\_init(): globus\_io\_tcp\_create\_listener() ...
- [Bug 3171](#):<sup>21</sup> add RFC 2253 principal name to JAAS subject?
- [Bug 3436](#):<sup>22</sup> globus\_uuid MAC address retrieval doesn't work on solaris
- [Bug 3509](#):<sup>23</sup> GLOBUS\_COMMON\_MODULE->GLOBUS\_CALLBACK\_MODULE deactivation...
- [Bug 3581](#):<sup>24</sup> gpt/globus\_core improvements
- [Bug 3610](#):<sup>25</sup> globus-makefile-header returns bad field
- [Bug 4035](#):<sup>26</sup> MacOS threading issues
- [Bug 4097](#):<sup>27</sup> Signal handling problem on MacOS X
- [Bug 4318](#):<sup>28</sup> Pkgdata needs to make source dir explicit

---

<sup>10</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=484](http://bugzilla.globus.org/globus/show_bug.cgi?id=484)

<sup>11</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=686](http://bugzilla.globus.org/globus/show_bug.cgi?id=686)

<sup>12</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=930](http://bugzilla.globus.org/globus/show_bug.cgi?id=930)

<sup>13</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=1141](http://bugzilla.globus.org/globus/show_bug.cgi?id=1141)

<sup>14</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2315](http://bugzilla.globus.org/globus/show_bug.cgi?id=2315)

<sup>15</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2791](http://bugzilla.globus.org/globus/show_bug.cgi?id=2791)

<sup>16</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2793](http://bugzilla.globus.org/globus/show_bug.cgi?id=2793)

<sup>17</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2794](http://bugzilla.globus.org/globus/show_bug.cgi?id=2794)

<sup>18</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2801](http://bugzilla.globus.org/globus/show_bug.cgi?id=2801)

<sup>19</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3081](http://bugzilla.globus.org/globus/show_bug.cgi?id=3081)

<sup>20</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3107](http://bugzilla.globus.org/globus/show_bug.cgi?id=3107)

<sup>21</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3171](http://bugzilla.globus.org/globus/show_bug.cgi?id=3171)

<sup>22</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3436](http://bugzilla.globus.org/globus/show_bug.cgi?id=3436)

<sup>23</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3509](http://bugzilla.globus.org/globus/show_bug.cgi?id=3509)

<sup>24</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3581](http://bugzilla.globus.org/globus/show_bug.cgi?id=3581)

<sup>25</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3610](http://bugzilla.globus.org/globus/show_bug.cgi?id=3610)

<sup>26</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4035](http://bugzilla.globus.org/globus/show_bug.cgi?id=4035)

<sup>27</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4097](http://bugzilla.globus.org/globus/show_bug.cgi?id=4097)

<sup>28</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4318](http://bugzilla.globus.org/globus/show_bug.cgi?id=4318)

## 5. For More Information

Click [here](#)<sup>29</sup> for more information about this component.

---

<sup>29</sup> [index.html](#)

---

# Chapter 5. GT 4.0.3 Incremental Release Notes: C Common Libraries

## 1. Introduction

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

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [C Common Libraries 4.0 Release Notes](#)<sup>1</sup>.

## 2. Changes Summary

No changes have occurred for C Common Libraries since GT 4.0.2.

## 3. Bug Fixes

No bugs were fixed in the C Common Libraries since the 4.0.2 release.

## 4. Known Problems

The following problems are known to exist for C Common Libraries at the time of the 4.0.3 release:

- [Bug 484](#):<sup>2</sup> Mem leak on deactivate\_all
- [Bug 686](#):<sup>3</sup> globus\_mutex\_lock() behaves differently on Unix -vs- Win32
- [Bug 930](#):<sup>4</sup> inttypes.h included without testing if it's present
- [Bug 1141](#):<sup>5</sup> Misleading error messages using globus-makefile-header
- [Bug 2315](#):<sup>6</sup> Globus common fails to compile if there is no ipv6 support
- [Bug 2791](#):<sup>7</sup> Command globus-domainname needs g11n
- [Bug 2793](#):<sup>8</sup> Command globus-generalized-time needs g11n
- [Bug 2794](#):<sup>9</sup> Command globus-hostname needs g11n

---

<sup>1</sup> [http://www.globus.org/toolkit/docs/4.0/common/ccommonlib/C\\_Common\\_Libraries\\_Release\\_Notes.html](http://www.globus.org/toolkit/docs/4.0/common/ccommonlib/C_Common_Libraries_Release_Notes.html)

<sup>2</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=484](http://bugzilla.globus.org/globus/show_bug.cgi?id=484)

<sup>3</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=686](http://bugzilla.globus.org/globus/show_bug.cgi?id=686)

<sup>4</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=930](http://bugzilla.globus.org/globus/show_bug.cgi?id=930)

<sup>5</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=1141](http://bugzilla.globus.org/globus/show_bug.cgi?id=1141)

<sup>6</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2315](http://bugzilla.globus.org/globus/show_bug.cgi?id=2315)

<sup>7</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2791](http://bugzilla.globus.org/globus/show_bug.cgi?id=2791)

<sup>8</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2793](http://bugzilla.globus.org/globus/show_bug.cgi?id=2793)

<sup>9</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2794](http://bugzilla.globus.org/globus/show_bug.cgi?id=2794)

- [Bug 2801](#):<sup>10</sup> Command globus-version needs g11n
- [Bug 3081](#):<sup>11</sup> Small leak in libltdl
- [Bug 3107](#):<sup>12</sup> Fatal error: tcp\_init(): globus\_io\_tcp\_create\_listener() ...
- [Bug 3171](#):<sup>13</sup> add RFC 2253 principal name to JAAS subject?
- [Bug 3436](#):<sup>14</sup> globus\_uuid MAC address retrieval doesn't work on solaris
- [Bug 3509](#):<sup>15</sup> GLOBUS\_COMMON\_MODULE->GLOBUS\_CALLBACK\_MODULE deactivation...
- [Bug 3581](#):<sup>16</sup> gpt/globus\_core improvements
- [Bug 3610](#):<sup>17</sup> globus-makefile-header returns bad field
- [Bug 4035](#):<sup>18</sup> MacOS threading issues
- [Bug 4097](#):<sup>19</sup> Signal handling problem on MacOS X
- [Bug 4318](#):<sup>20</sup> Pkgdata needs to make source dir explicit

## 5. For More Information

Click [here](#)<sup>21</sup> for more information about this component.

---

<sup>10</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2801](http://bugzilla.globus.org/globus/show_bug.cgi?id=2801)

<sup>11</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3081](http://bugzilla.globus.org/globus/show_bug.cgi?id=3081)

<sup>12</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3107](http://bugzilla.globus.org/globus/show_bug.cgi?id=3107)

<sup>13</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3171](http://bugzilla.globus.org/globus/show_bug.cgi?id=3171)

<sup>14</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3436](http://bugzilla.globus.org/globus/show_bug.cgi?id=3436)

<sup>15</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3509](http://bugzilla.globus.org/globus/show_bug.cgi?id=3509)

<sup>16</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3581](http://bugzilla.globus.org/globus/show_bug.cgi?id=3581)

<sup>17</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3610](http://bugzilla.globus.org/globus/show_bug.cgi?id=3610)

<sup>18</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4035](http://bugzilla.globus.org/globus/show_bug.cgi?id=4035)

<sup>19</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4097](http://bugzilla.globus.org/globus/show_bug.cgi?id=4097)

<sup>20</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4318](http://bugzilla.globus.org/globus/show_bug.cgi?id=4318)

<sup>21</sup> [index.html](#)

---

# Chapter 6. GT 4.0.4 Incremental Release Notes: C Common Libraries

## 1. Introduction

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

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [C Common Libraries 4.0 Release Notes](#)<sup>1</sup>.

## 2. Changes Summary

No changes have been made for the C common libraries since 4.0.3 other than the listed bugfixes.

## 3. Bug Fixes

- [Bug 4585](#):<sup>2</sup> Need to allow intel cc (icc) for x86\_64 as a vendorcc flavor
- [Bug 4586](#):<sup>3</sup> int64\_t not defined on Suse9/x86\_64 for non-gcc compilers
- [Bug 4942](#):<sup>4</sup> Workaround getnameinfo bug in OS X on Intel
- [Bug 4966](#):<sup>5</sup> Crash in globus\_uuid\_create on OS X x86\_64

## 4. Known Problems

- [Bug 484](#):<sup>6</sup> Mem leak on deactivate\_all
- [Bug 686](#):<sup>7</sup> globus\_mutex\_lock() behaves differently on Unix -vs- Win32
- [Bug 930](#):<sup>8</sup> inttypes.h included without testing if it's present
- [Bug 1141](#):<sup>9</sup> Misleading error messages using globus-makefile-header
- [Bug 2791](#):<sup>10</sup> Command globus-domainname needs g11n
- [Bug 2793](#):<sup>11</sup> Command globus-generalized-time needs g11n

---

<sup>1</sup> [http://www.globus.org/toolkit/docs/4.0/common/ccommonlib/C\\_Common\\_Libraries\\_Release\\_Notes.html](http://www.globus.org/toolkit/docs/4.0/common/ccommonlib/C_Common_Libraries_Release_Notes.html)

<sup>2</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4585](http://bugzilla.globus.org/globus/show_bug.cgi?id=4585)

<sup>3</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4586](http://bugzilla.globus.org/globus/show_bug.cgi?id=4586)

<sup>4</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4942](http://bugzilla.globus.org/globus/show_bug.cgi?id=4942)

<sup>5</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4966](http://bugzilla.globus.org/globus/show_bug.cgi?id=4966)

<sup>6</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=484](http://bugzilla.globus.org/globus/show_bug.cgi?id=484)

<sup>7</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=686](http://bugzilla.globus.org/globus/show_bug.cgi?id=686)

<sup>8</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=930](http://bugzilla.globus.org/globus/show_bug.cgi?id=930)

<sup>9</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=1141](http://bugzilla.globus.org/globus/show_bug.cgi?id=1141)

<sup>10</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2791](http://bugzilla.globus.org/globus/show_bug.cgi?id=2791)

<sup>11</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2793](http://bugzilla.globus.org/globus/show_bug.cgi?id=2793)

- [Bug 2794](#):<sup>12</sup> Command globus-hostname needs g11n
- [Bug 3081](#):<sup>13</sup> Small leak in libltdl
- [Bug 3107](#):<sup>14</sup> Fatal error: tcp\_init(): globus\_io\_tcp\_create\_listener() ...
- [Bug 3436](#):<sup>15</sup> globus\_uuid MAC address retrieval doesn't work on solaris
- [Bug 3509](#):<sup>16</sup> GLOBUS\_COMMON\_MODULE->GLOBUS\_CALLBACK\_MODULE deactivation...
- [Bug 3610](#):<sup>17</sup> globus-makefile-header returns bad field
- [Bug 4035](#):<sup>18</sup> MacOS threading issues
- [Bug 4097](#):<sup>19</sup> Signal handling problem on MacOS X
- [Bug 4318](#):<sup>20</sup> Pkgdata needs to make source dir explicit
- [Bug 4348](#):<sup>21</sup> globus-makefile-header generates invalid makefile for Index Service bindings

## 5. For More Information

Click [here](#):<sup>22</sup> for more information about this component.

---

<sup>12</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=2794](http://bugzilla.globus.org/globus/show_bug.cgi?id=2794)

<sup>13</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3081](http://bugzilla.globus.org/globus/show_bug.cgi?id=3081)

<sup>14</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3107](http://bugzilla.globus.org/globus/show_bug.cgi?id=3107)

<sup>15</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3436](http://bugzilla.globus.org/globus/show_bug.cgi?id=3436)

<sup>16</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3509](http://bugzilla.globus.org/globus/show_bug.cgi?id=3509)

<sup>17</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3610](http://bugzilla.globus.org/globus/show_bug.cgi?id=3610)

<sup>18</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4035](http://bugzilla.globus.org/globus/show_bug.cgi?id=4035)

<sup>19</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4097](http://bugzilla.globus.org/globus/show_bug.cgi?id=4097)

<sup>20</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4318](http://bugzilla.globus.org/globus/show_bug.cgi?id=4318)

<sup>21</sup> [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=4348](http://bugzilla.globus.org/globus/show_bug.cgi?id=4348)

<sup>22</sup> [index.html](#)

---

# Chapter 7. GT 4.0 C Common Libraries : System Administrator's Guide

## 1. Introduction

This guide contains advanced configuration information for system administrators working with C Common Libraries. It provides references to information on procedures typically performed by system administrators, including installation, configuring, deploying, and testing the installation.

### Important

This information is in addition to the basic Globus Toolkit prerequisite, overview, installation, security configuration instructions in the [GT 4.0 System Administrator's Guide](#)<sup>1</sup>. Read through this guide before continuing!

## 2. Building and Installing

For instructions on building and installing GT 4.0, see the [Installation Guide](#)<sup>2</sup>.

## 3. Configuring

### 3.1. Configuration overview

There is no configuration interface to the C common libraries.

## 4. Deploying

The C common libraries are available in any GT 4.0 installation.

## 5. Testing

The C common libraries test suite is included in the `globus_common_test` package.

## 6. Security Considerations

There are no security considerations for the C common libraries.

## 7. Troubleshooting

The C common libraries are intended for use by developers; there are no user level troubleshooting techniques.

---

<sup>1</sup> `../admin/docbook/`

<sup>2</sup> <http://www.globus.org/toolkit/docs/4.0/admin/>

---

# Chapter 8. GT 4.0 C Common Libraries : User's Guide

## 1. 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.

## 2. Command-line tools

There are no command line interfaces to the C common libraries.

## 3. Troubleshooting

The C common libraries are intended for use by developers; there are no user level troubleshooting techniques.

---

# Chapter 9. GT 4.0 C Common Libraries: Developer's Guide

## 1. 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.

## 2. Before you begin

### 2.1. Feature summary

Features new in release GT 4.0

- `globus_range_list` abstraction added
- `globus_logging` abstraction added

### 2.2. Tested platforms

The C common libraries work on any platform on which the toolkit is supported. See [supported platforms](#)<sup>1</sup>.

### 2.3. Backward compatibility summary

API changes since GT version 3.2

- `globus_range_list` abstraction added
- `globus_logging` abstraction added

All of the GT 3.2 API is still functional in GT 4.0

### 2.4. Technology dependencies

C Common Libraries only depend on the `globus_core` module.

### 2.5. Security considerations

There are no security considerations for the C common libraries.

## 3. Architecture and design overview

Not available at this time.

---

<sup>1</sup> <http://www.globus.org/toolkit/docs/4.0/admin/docbook/ch03.html#s-platform>

## 4. Public interface

The semantics and syntax of the APIs for this component can be found in the [public interface guide](#)<sup>2</sup>.

## 5. 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.

## 6. Debugging

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

## 7. Troubleshooting

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

## 8. Related Documentation

See the [C API pages](#)<sup>3</sup> for more information about this component.

---

<sup>2</sup> [C\\_Common\\_Libraries\\_Public\\_Interfaces.html](#)

<sup>3</sup> <http://www.globus.org/api/c-globus-4.0/>

---

# Chapter 10. GT 4.0 Component Fact Sheet: C Common Libraries

## 1. Brief 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. Summary of features

Features new in release GT 4.0

- globus\_range\_list abstraction added
- globus\_logging abstraction added

## 3. Usability summary

There are no usability issues with the C common libraries as they are only intended for development.

## 4. Backward compatibility summary

API changes since GT version 3.2

- globus\_range\_list abstraction added
- globus\_logging abstraction added

All of the GT 3.2 API is still functional in GT 4.0

## 5. Technology dependencies

C Common Libraries only depend on the globus\_core module.

## 6. Tested platforms

The C common libraries work on any platform on which the toolkit is supported. See [supported platforms](#)<sup>1</sup>.

## 7. Associated standards

There are no standards implemented by the C common libraries.

---

<sup>1</sup> <http://www.globus.org/toolkit/docs/4.0/admin/docbook/ch03.html#s-platform>

## 8. For More Information

See the [C API pages](#)<sup>2</sup> for more information about this component.

---

<sup>2</sup> <http://www.globus.org/api/c-globus-4.0/>

---

# Chapter 11. GT 4.0 Component Guide to Public Interfaces: C Common Libraries

## 1. Semantics and syntax of APIs

### 1.1. Component API

See the [C API pages](#)<sup>1</sup> for other API documentation on `globus_common`.

### 1.2. Internationalization Infrastructure

The Globus Toolkit C Common Library now has optional infrastructure support for internationalization, which is used by GridFTP and its dependencies (pre-ws authorization/authentication and XIO).

This means that user-presented strings are wrapped in a lookup function, which, if the `globus_i18n` module is installed and the `GLOBUS_I18N` environment variable is set to "YES", will lookup the string in a resource bundle using ICU4C. If `GLOBUS_I18N` is set to "NO", or the `globus_i18n` module is not installed, or the string value cannot be found in the resource bundle, the default string (exactly what was being looked up) is returned.

### 1.3. Developer Information

There are two functions that are used for string lookup.

The first is the preferred function; you supply a module descriptor and the string that you want to look up:

```
globus_common_i18n_get_string(  
    globus_module_descriptor_t * module,  
    char * key);
```

The second function is used if you need to look up a particular locale. A NULL value for locale will look up from the default locale:

```
globus_common_i18n_get_string_by_key(  
    char * locale,  
    char * resource_name,  
    char * key);
```

Typically, one or more macros will be defined on a per-module basis that supply the module descriptor, to reduce clutter in the code. Within the toolkit, these are typically `___SL` where `___` are some mnemonic for the module in question (for example `_GCSL` is defined for `globus_common`).

The resource bundles used for the string lookups are created using ICU4C (see [IBM documentation on Resource Bundles](#))<sup>2</sup>.

---

<sup>1</sup> <http://www.globus.org/api/c-globus-4.0/>

<sup>2</sup> <http://oss.software.ibm.com/icu/userguide/ResourceManagement.html>

Our resource bundles are very simple; they contain simply a set of keys and strings. The key is actually the string itself: it is hashed using the `globus_hashtable_string_hash` function, then converted to contain only invariant characters ( `#!@[]^{}~` are converted to `'_'`). See `globus_i18n_resource_init.c` in the `globus_i18n` source for an example of creating keys.

While resource bundles for the Globus Toolkit are not by default created as part of the build process nor distributed in our binary distributions, there is a tool distributed with the `globus_i18n` package that makes them simple to construct.

Invoking `globus-i18n-resource-create <module name>` from the top level directory of a built (or at least configured) source package will generate a resource bundle for that package which can then be moved to `$GLOBUS_LOCATION/share/i18n/`.

`globus-i18n-resource-create` is part of the `globus_i18n` package, and will be installed if `--enable-i18n` was given as a configure option to the installer. `globus-i18n-resource-create` uses `globus-i18n-resource-init` and `genrb` from ICU4C to create the resource bundles.

Resource bundles must be installed in `$GLOBUS_LOCATION/share/i18n/`.

## 2. Semantics and syntax of the WSDL

There is no support for this type of interface.

## 3. Command-line tools

There are no command line interfaces to the C common libraries.

## 4. Overview of Graphical User Interface

There is no support for this type of interface.

## 5. Semantics and syntax of domain-specific interface

There is no support for this type of interface.

## 6. Configuration interface

### 6.1. Configuration overview

There is no configuration interface to the C common libraries.

## 7. Environment variable interface

- `GLOBUS_ERROR_VERBOSE=1` can be set to enable verbose error messages.
- `GLOBUS_ERROR_OUTPUT=1` can be set to enable output of all errors (including some that should be ignored).

---

# Chapter 12. GT 4.0 C Common Libraries: Quality Profile

## 1. Test coverage reports

There are no reports on this component.

## 2. Code analysis reports

There are no reports on this component.

## 3. Outstanding bugs

See a [bugzilla](http://bugzilla.globus.org/globus/query.cgi)<sup>1</sup> query on Toolkit Internals->globus\_common to list the bugs outstanding.

## 4. Bug Fixes

See a [bugzilla](http://bugzilla.globus.org/globus/query.cgi)<sup>2</sup> query on Toolkit Internals->globus\_common to list the bugs fixed.

## 5. Performance reports

There are no reports on this component.

---

<sup>1</sup> <http://bugzilla.globus.org/globus/query.cgi>

<sup>2</sup> <http://bugzilla.globus.org/globus/query.cgi>

---

# Chapter 13. GT 4.0 Migrating Guide for C Common Libraries

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

## 1. Migrating from GT2

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

## 2. Migrating from GT3

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