

SAGA C++

Ole Weidner, Center for Computation & Technology

Version: 0.1

June 5, 2011

SAGA C++ Adaptor Guide

Abstract

This documents provides an overview of all available Adaptors for the SAGA C++ Reference Implementation.

Copyright Notice

Copyright © CCT / Louisiana State Univeristy(2007). All Rights Reserved.

Contents

1	Globus GridFTP File Adaptor	3
1.1	Overview	3
1.2	Details	3
1.3	Configuration	3
1.4	Additional Notes	3
1.5	Capabilites	4
2	Globus GRAM2 Job Adaptor	6
2.1	Overview	6
2.2	Details	6
2.3	Configuration	6
2.4	Additional Notes	6
2.5	Capabilites	6
3	Writing new SAGA adaptors	7
3.1	Overview	7
	References	9

1 Globus GridFTP File Adaptor

1.1 Overview

Date: 2008/01/01

Author: Ole Weidner (oweidner@cct.lsu.edu)

The Globus GridFTP Adaptor implements the SAGA filesystem API for interaction with local and remote filesystem using the Globus GridFTP protocol.

1.2 Details

Library Name: `libsaga_adaptor_globus_gridftp_file.[a so dylib]`

Configuration File: `globus_gridftp_file_adaptor.ini`

Accepted URL Schemes: `any:// gridftp:// gsiftp://`

Accepted Copy/Move Targets: `any:// file:// gridftp:// gsiftp://`

1.3 Configuration

This adaptor doesn't have any configurable options.

1.4 Additional Notes

- The adaptor requires a valid Globus X.509 proxy certificate.
- This adaptor cannot be used with old GridFTP servers prior to version 1.17 (Globus Toolkit 3.2) since some required protocol functionality is not available.

1.5 Capabilities

Namespace Entry API Methods	directory	file
<code>ns_entry::get_url ()</code>	✓	✓
<code>ns_entry::get_cwd ()</code>	✓	✓
<code>ns_entry::get_name ()</code>	✓	✓
<code>ns_entry::is_dir ()</code>	✓	✓
<code>ns_entry::is_entry ()</code>	✓	✓
<code>ns_entry::is_link ()</code>	✓	✓
<code>ns_entry::read_link ()</code>	✓	✓
<code>ns_entry::copy (target)</code>	no ²	✓
<code>ns_entry::link (target)</code>	no ¹	no ¹
<code>ns_entry::move (target)</code>	✓	✓
<code>ns_entry::remove ()</code>	✓	✓
<code>ns_entry::close ()</code>	✓	✓
<code>ns_entry::permission_allow (perm)</code>	no ²	no ²
<code>ns_entry::permission_deny (perm)</code>	no ²	no ²

Filesystem File API Methods	file
<code>file::get_size ()</code>	✓
<code>file::read ()</code>	✓
<code>file::write ()</code>	✓
<code>file::seek (name)</code>	✓
<code>file::read_p (pattern)</code>	no ¹
<code>file::write_p (pattern)</code>	no ¹
<code>file::size_p (pattern)</code>	no ¹
<code>file::modes_e (emode)</code>	no ¹
<code>file::write_e (emode)</code>	no ¹
<code>file::write_p (emode)</code>	no ¹
<code>file::write_p (emode)</code>	no ¹

Namespace Directory API Methods	directory
<code>ns_dir::find (pattern)</code>	no ²
<code>ns_dir::read_link (name)</code>	
<code>ns_dir::exists (name)</code>	
<code>ns_dir::is_dir (name)</code>	✓
<code>ns_dir::is_entry (name)</code>	✓
<code>ns_dir::is_link (name)</code>	✓
<code>ns_dir::copy (src, target)</code>	✓
<code>ns_dir::link (src, target)</code>	no ¹
<code>ns_dir::move (src, target)</code>	✓
<code>ns_dir::remove (target)</code>	✓
<code>ns_dir::make_dir (name)</code>	
<code>ns_dir::open (name)</code>	
<code>ns_dir::open_dir (name)</code>	
<code>ns_dir::get_num_entries ()</code>	no ²
<code>ns_dir::get_entry (index)</code>	no ²
<code>ns_dir::permission.allow</code>	no ²
<code>ns_dir::permission.deny</code>	no ²
Filesystem Directory API Methods	directory
<code>directory::get_size ()</code>	✓
<code>directory::is_file (name)</code>	✓
<code>directory::open_dir (name)</code>	✓
<code>directory::open (name)</code>	✓

¹Not supported by the GridFTP protocol.²Not implemented yet.

2 Globus GRAM2 Job Adaptor

2.1 Overview

Date: 2008/01/01

Author: Ole Weidner (oweidner@cct.lsu.edu)

The Globus GRAM2 Adaptor implements the SAGA job API for interaction with GRAM2 (pre-WS) resource managers.

2.2 Details

Library Name: libsaga_adaptor_globus_gram_job.[a so dylib]

Configuration File: globus_gram_job_adaptor.ini

Accepted URL Schemes: any:// gram://

2.3 Configuration

This adaptor doesn't have any configurable options.

2.4 Additional Notes

The adaptor requires a valid Globus X.509 proxy certificate.

2.5 Capabilites

3 Writing new SAGA adaptors

3.1 Overview

The SAGA C++ sources include an adaptor generator, which allows to easily create stubs for custom adaptors. The script is located in

`adaptors/generator/generator.pl`

and is installed into `\$SAGA_LOCATION/bin/`. Calling that script without any arguments will print a help screen, which provides a number of details on the command line arguments etc.

The exemplary shell session shown below demonstrates the use of the adaptor generator, and results in a complete file adaptor:

```
# cd saga-core-src/adaptors/

# ./generator/saga-adaptor-generator.pl -s ssh -n sshfs -t file -d .

suite:          ssh
type:           file
ftype:          sshfs_file
name:           sshfs
directory:      ./ssh/ssh_sshfs_file

copying files:   ...
fixing file names: .....
fixing files:    .....

You can now cd to ./ssh/ssh_sshfs_file,
and run 'make; make install'.

Note that you need to set SAGA_LOCATION before,
and point it to your SAGA installation tree.

# cd ssh/ssh_sshfs_file

# make
compiling  ssh_sshfs_file_adaptor.o
compiling  ssh_sshfs_file_dir_impl.o
compiling  ssh_sshfs_file_dir_nsdire_impl.o
compiling  ssh_sshfs_file_dir_nsentry_impl.o
compiling  ssh_sshfs_file_dir_perm_impl.o
compiling  ssh_sshfs_file_file_impl.o
compiling  ssh_sshfs_file_file_nsentry_impl.o
compiling  ssh_sshfs_file_file_perm_impl.o
```

```
liblinking    libsaga_adaptor_ssh_sshfs_file.so
liblinking    libsaga_adaptor_ssh_sshfs_file.a (static)

# make install
installing    lib
installing    lib (static)
installing    adaptor ini
```

When running any SAGA application, that adaptor will get loaded and will receive requests to perform remote operations. That can be confirmed by setting `SAGA_VERBOSE` in the application environment (see SAGA installation guide).

Of course, that adaptor will not be able to perform any meaningful operation – it is just a stub, and will simply throw `NotImplemented` exceptions for all calls. However, it is now straight forward to fill the stub with the respective functionality.

FIXME: Need to add details about class hierarchy, adaptor data, and instance data. Also, some details about adaptor makefiles, configure support etc might be useful.

Note that each adaptor will end up in a separate shared library. Since a typical SAGA installation will use multiple adaptors, and thus multiple adaptor libraries will be loaded into the same address space, the adaptor programmer needs to make sure to choose unique symbol names (i.e. to choose a unique and descriptive symbol name space), to avoid runtime symbol clashes.

Appendix