GT 4.0 Contribution: pyGridWare

GT 4.0 Contribution: pyGridWare

Table of Contents

1. Key Concepts 1
2. 4.0.0 Release Notes 2
    1. Component Overview 2
    2. Feature Summary 2
    3. Technology Dependencies 2
    4. Supported Platforms 3
    5. Backward Compatibility Summary 3
    6. For More Information 3
3. 4.0.1 Release Notes 4
    1. Introduction 4
    2. Changes Summary 4
    3. Bug Fixes 4
    4. Known Problems 4
    5. For More Information 4
4. 4.0.2 Release Notes 5
    1. Introduction 5
    2. Changes Summary 5
    3. Bug Fixes 5
    4. Known Problems 5
    5. For More Information 6
5. 4.0.3 Release Notes 7
    1. Introduction 7
    2. Changes Summary 7
    3. Bug Fixes 7
    4. Known Problems 7
    5. For More Information 7
6. 4.0.4 Release Notes 8
    1. Introduction 8
    2. Changes Summary 8
    3. Bug Fixes 8
    4. Known Problems 8
    5. For More Information 8
7. System Administrator's Guide 9
    1. Introduction 9
    2. Dependencies 9
    3. Building and installing 9
    4. Configuring 10
    5. Deploying 10
    6. Testing 11
    7. Security considerations 11
    8. Troubleshooting 12
8. Fact Sheet 13
    1. Brief component overview 13
    2. Summary of features 13
    3. Backward compatibility summary 13
    4. Technology dependencies 13
    5. Tested platforms 14
    6. Associated standards 14
    7. For More Information 14

Chapter 1. Key Concepts

See the key concepts [1] from Common Runtime Components.

[1] http://www.globus.org/toolkit/docs/4.0/common/key

Chapter 2. GT 4.0 Release Notes: pyGridWare

1. Component Overview

pyGridWare, the python WS Core, provides a basic python toolset for creating WSRF enabled web services, proven to interoperate with the Java WSRF. Performance is a primary concern and motivation. This WSRF project realizes a significant speedup by using C implementations at the performance critical security and DOM layers.

WSRF support includes WS-Resource Lifetime, WS-Resource Properties, and WS-Notification. Updated security support for WS-Secure Conversation and Secure Message is also included.

Client and service stubs have been generated from the most recent OASIS drafts and WS-Secure Conversation specification, but the bindings can be easily regenerated from WSDL via the setup script at install whenever WSDL needs to be updated.

2. Feature Summary

Features new in release GT 4.0:

- [list features]

Other Supported Features

- [list features]

Deprecated Features

- [list features]

3. Technology Dependencies

Required (Use newest version if possible):

- python 2.3[1]
- pyXML-0.8.4[2]
- 4Suite-1.0a4[3]
- Twisted-1.3.0[4]

Optional: WS-Security: XML Digital Signatures, Secure Conversation, etc.

- pyGlobus[5] Python Wrapper for GT2, needs security and utility modules.

[1] http://www.python.org
[2] http://pyxml.sourceforge.net/
[3] http://sourceforge.net/projects/foursuite/
[4] http://www.twistedmatrix.com/products/download
[5] http://dsd.lbl.gov/gtg/projects/pyGlobus/FAQ.html

- Globus 2 Security and utility packages.[6]

4. Supported Platforms

Tested Platforms for pyGridWare:

- [list platforms for which component has been tested]

5. Backward Compatibility Summary

This implementation will not be backward compatible with any previous GT3 release.

6. For More Information

Click here[7] for more information about this component.

[6] http://www.globus.org
[7] index.html

Chapter 3. GT 4.0.1 Incremental Release Notes: Python WS Core

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 Python WS Core 4.0 Release Notes[1].

2. Changes Summary

The following changes have occurred for Python WS Core:

- Added "defer" attribute to generated client bindings. If set to "True" method invocations return a twisted deferred instance, else if set to "False" (default) they return an instance representing the response message. Thus it's easy to switch between asynchronous and synchronous programming paradigms.
- New resources can be dropped into the service container at runtime as twisted "rpy" scripts. So resources (eg. CounterService) are added or removed from the service container by simply adding or removing a rpy script (eg. CounterService.rpy).
- WS-BaseNotification, added support for pausing and resuming subscriptions through the Subscription Manager.
- WS-ResourceProperties operations QueryResourceProperties and GetMultipleResourceProperties are now supported/implemented in generated service stubs.

3. Bug Fixes

No bugs were fixed for Python WS Core.

4. Known Problems

No problems are known to exist for Python WS Core at the time of the 4.0.1 release.

5. For More Information

Click here[2] for more information about this component.

[1] http://www.globus.org/toolkit/docs/4.0/contributions/pythonwscore/Python_WS_Core_Release_Notes.html
[2] index.html

Chapter 4. GT 4.0.2 Incremental Release Notes: Python WS Core

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 Python WS Core 4.0 Release Notes[1].

2. Changes Summary

The following changes have occurred for Python WS Core since 4.0.1:

- Various performance modifications, approximately halving the time for a single rpc operation.
- Added python "proxy" tools for creating and managing a RFC 3820 compliant proxy.
- Complete separation between the pyGridWare installation and your site.
- New drop-in services based on twisted "rpy" scripts.
- New tool "wsdl2web.py" that creates a new web/grid services site from a WSDL definition, so one can move quickly from WSDL authoring to deployment.
- Additional "layer" on top of soap stack, making it easier to use ZSI python objects.
- Additional support for M2Crypto, which is replacing pyGlobus.security as the crypto package for message layer security.
- GRAM client "pyGridWare/bin/ManagedJobClient.py"
- New Tutorial at http://dsd.lbl.gov/gtg/projects/pyGridWare/doc/tutorial/html/index.html.

3. Bug Fixes

Many bugs have been fixed for Python WS Core in the ZSI SOAP layer since 4.0.1. Check out the ZSI bug tracker on sourceforge.

4. Known Problems

Message layer security in Python WS Core is in a state of flux, the release of 4.0.2 came as a surprise and we were in the middle of a move. The client- side is working with globus-4.0.2rc1.

[1] http://www.globus.org/toolkit/docs/4.0/contributions/pythonwscore/Python_WS_Core_Release_Notes.html

5. For More Information

Click here[2] for more information about this component.

[2] index.html

Chapter 5. GT 4.0.3 Incremental Release Notes: Python WS Core

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 Python WS Core 4.0 Release Notes[1].

2. Changes Summary

Other than bug fixes, the only change that has occurred for Python WS Core since 4.0.2 is the fixing of a potential proxy generation race condition bug in "pyGsi", similar to the one in the Globus Toolkit

3. Bug Fixes

Many bugs have been fixed for Python WS Core in the ZSI SOAP layer since 4.0.2. Check out the ZSI bug tracker on SourceForge[2].

4. Known Problems

No problems are known to exist for Python WS Core at the time of the 4.0.3 release.

5. For More Information

Click here[3] for more information about this component.

[1] http://www.globus.org/toolkit/docs/4.0/contributions/pythonwscore/Python_WS_Core_Release_Notes.html
[2] http://sourceforge.net/tracker/?atid=387667&group_id=26590&func=browse
[3] index.html

Chapter 6. GT 4.0.4 Incremental Release Notes: Python WS Core

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 Python WS Core 4.0 Release Notes[1].

2. Changes Summary

- Authorization portal framework for controlling resource and service instance level access.
- Expanded unit test coverage of SSL and GSI authentication/authorization mechanisms.
- "samples/globus" directory of sample client scripts for using various gt4 services.
- New WS Delegation client library (pyGridWare.common.delegationlib).
- New Reliable File Transfer client library (pyGridWare.common.rftlib).
- Script wsdl2web, all logic is now in a module (pyGridWare.utility.generate.wsdl2web).

3. Bug Fixes

- Added substitution group support

4. Known Problems

For information on existing bugs, click here[2].

5. For More Information

Click here[3] for more information about this component.

[1] http://www.globus.org/toolkit/docs/4.0/contributions/pythonwscore/Python_WS_Core_Release_Notes.html
[2] http://bugzilla.globus.org/globus/buglist.cgi?product=Python%20Grid%20Tools&bug_status=NEW&bug_status=ASSIGNED&bug_status=REOPEND
[3] http://dev.globus.org/wiki/Python_Core

Chapter 7. GT 4.0 pyGridWare Contribution: System Administrator's Guide

1. Introduction

This guide contains advanced configuration information for system administrators working with pyGridWare, the python WS-Core. 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[1]. Read through this guide before continuing!

2. Dependencies

Required (Use newest version if possible):

- python 2.3[2]
- pyXML-0.8.4[3]
- 4Suite-1.0a4[4]
- Twisted-1.3.0[5]

Optional: WS-Security: XML Digital Signatures, Secure Conversation, etc.

- pyGlobus[6] Python Wrapper for GT2, needs security and utility modules.
- Globus 2 Security and utility packages.[7]

3. Building and installing

Untar the tarball or grab the CVS source, run pyGridWare distutils setup script, optionally regenerate all bindings.

1. Do one of the following:
2. Untar tarball:

```
%tar zxf pyGridWare-1.04b.tar.gz
```

[1] ../../admin/docbook/
[2] http://www.python.org
[3] http://pyxml.sourceforge.net/
[4] http://sourceforge.net/projects/foursuite/
[5] http://www.twistedmatrix.com/products/download
[6] http://dsd.lbl.gov/gtg/projects/pyGlobus/FAQ.html
[7] http://www.globus.org

3. Grab the CVS source:

```
%cvs -d :pserver:anonymous@bosshog.lbl.gov:/home/portnoy/u5/repository co login
%cvs -d :pserver:anonymous@bosshog.lbl.gov:/home/portnoy/u5/repository co pyGridWare
```

4. Change directory

```
%cd pyGridWare
```

5. Run distutils

```
%python setup.py --regenerate install
```

If you want Message Level Security, build the GT2 libraries with a threaded flavor. You can install any GT3 All Source installer bundles. Directions for installing GT 3.2 can be found here[8].

To build pyGlobus:

1. Set the environment variables *GLOBUS_LOCATION* and *GPT_LOCATION* to the location of the globus installation.

2. Run the pyGlobus distutils setup script:

```
%python setup.py install
```

4. Configuring

There are two configuration files:

4.1. pyGridWare/config.txt

Read by the distutils setup script when "regenerate" is specified. This is where all service WSDLs are specified. When setup is run, bindings will be created automatically.

```
[WSDL]
CounterService = share/schema/core/samples/counter/counter_service.wsdl
```

4.2. pyGridWare/bin/config.txt

Configuration parameters for logging, security, container location, and service paths. This file needs to be in the directory from where you are executing the client and/or server.

5. Deploying

Run the container script

[8] http://www.globus.org/toolkit/docs/3.2/installation/install_installing.html

1. Change to bin directory

   ```
   %cd bin
   ```

2. Edit the file `server-config.tac` by adding or removing resources:

   ```
   def GetResource(contextFactory=None):
     root = Resource()
     root.putChild('wsrf', Resource())

     resource = Resource()
     root.getStaticEntity('wsrf').putChild('services', resource)

     resource.putChild('CounterService', Counter(post='/wsrf/services/CounterService'))
   ```

3. Run the start container script:

   ```
   %./start-container.sh
   ```

6. Testing

To test your installation, run:

```
%python unittest
```

7. Security considerations

Individual services can be configured with or without message security, but transport security is a characteristic of the entire container (either using ssl or plain tcp). Authentication and authorization of clients is performed using a callback mechanism.

7.1. Transport Security

Simply edit the file `config.txt` where the executable is being run and turn on ssl.

By default, pyGridWare will look in the user's home directory for the `.globus/usercert.pem` and `.globus/user-key.pem` files.

To use the grid proxy generated by `grid-proxy-init`, just specify the `/tmp/x509***` as the certfile and keyfile.

Example pyGridWare/bin/config.txt

```
[security]
ssl = 1
certfile =
keyfile =
```

8. Troubleshooting

1. Make sure all dependencies are met.

2. Building and installing problems:

3. python-2.3 problems We have recently discovered a problem running the distutils `--regenerate` that is caused by a bug in "urllib.basejoin". This is fixed in python2.4. ZSI needs to be patched to fix this problem in python-2.3.

Chapter 8. GT 4.0 Component Fact Sheet: pyGridWare (Contribution)

1. Brief component overview

pyGridWare, the python WS Core, provides a basic python toolset for creating WSRF enabled web services, proven to interoperate with the Java WSRF. Performance is a primary concern and motivation. This WSRF project realizes a significant speedup by using C implementations at the performance critical security and DOM layers.

WSRF support includes WS-Resource Lifetime, WS-Resource Properties, and WS-Notification. Updated security support for WS-Secure Conversation and Secure Message is also included.

Client and service stubs have been generated from the most recent OASIS drafts and WS-Secure Conversation specification, but the bindings can be easily regenerated from WSDL via the setup script at install whenever WSDL needs to be updated.

2. Summary of features

Features new in release GT 4.0:

- [list features]

Other Supported Features

- [list features]

Deprecated Features

- [list features]

3. Backward compatibility summary

This implementation will not be backward compatible with any previous GT3 release.

4. Technology dependencies

Required (Use newest version if possible):

- python 2.3[1]
- pyXML-0.8.4[2]
- 4Suite-1.0a4[3]
- Twisted-1.3.0[4]

[1] http://www.python.org
[2] http://pyxml.sourceforge.net/
[3] http://sourceforge.net/projects/foursuite/
[4] http://www.twistedmatrix.com/products/download

Optional: WS-Security: XML Digital Signatures, Secure Conversation, etc.

- pyGlobus[5] Python Wrapper for GT2, needs security and utility modules.
- Globus 2 Security and utility packages.[6]

5. Tested platforms

Tested Platforms for pyGridWare:

- [list platforms for which component has been tested]

6. Associated standards

Associated standards for Python WS Core:

- standard #1
- ...
- standard #n

7. For More Information

Click here[7] for more information about this component.

[5] http://dsd.lbl.gov/gtg/projects/pyGlobus/FAQ.html
[6] http://www.globus.org
[7] index.html