

Hawkeye Information Provider

Hawkeye Information Provider

Abstract

The Hawkeye Information Provider gathers Hawkeye data about Condor pool resources using the XML mapping of the GLUE schema and reports it to a GRAM4 service, which publishes it as resource properties. The official Condor site states the following:

Hawkeye utilizes the technologies already present in Condor and ClassAds to provide rich mechanisms for collecting, storing, and using information about computers. A Hawkeye system can be used to monitor various attributes of a collection of systems. The monitoring mechanism may also be used then to further the management of systems.

This information includes:

- basic host data (name, ID)
- processor information
- memory size
- OS name and version
- file system data
- processor load data
- other basic Condor host data

You can download a PDF version of Hawkeye Information Provider information [here](#)¹.

¹ [hawkeye.pdf](#)

Table of Contents

1. Release Notes	1
1. Component Overview	1
2. Feature Summary	1
3. Changes Summary	1
4. Bug Fixes	1
5. Known Problems	1
6. Technology Dependencies	2
7. Tested Platforms	2
8. Backward Compatibility Summary	2
9. Associated Standards	2
10. For More Information	2
2. Reference Guide	3
1. Overview	3
2. Prerequisites	3
3. Configuring	3
4. Resource Properties	4
5. Schema	4
6. Security Considerations	4
7. Testing	4
8. Troubleshooting	5
Glossary	6

Chapter 1. GT 4.2.1 Release Notes: Hawkeye Info Information Provider

1. Component Overview

The Hawkeye Information Provider gathers Hawkeye data about Condor pool resources using the XML mapping of the GLUE schema and reports it to a GRAM4 service, which publishes it as resource properties. The official Condor site states the following:

Hawkeye utilizes the technologies already present in Condor and ClassAds to provide rich mechanisms for collecting, storing, and using information about computers. A Hawkeye system can be used to monitor various attributes of a collection of systems. The monitoring mechanism may also be used then to further the management of systems.

This information includes:

- basic host data (name, ID)
- processor information
- memory size
- OS name and version
- file system data
- processor load data
- other basic Condor host data

2. Feature Summary

Features new in release 4.2.1:

- This provider publishes information such as the following: basic host data (name, ID), processor information, memory size, OS name and version, file system data, processor load data, other basic Condor host data.

3. Changes Summary

This is an existing information provider available from GT4 to GT 4.2.1.

4. Bug Fixes

There are ongoing fixed bugs for this information provider (see Bugzilla).

5. Known Problems

- There may currently be bugs for this information provider (see [Bugzilla](#)¹).

¹ <http://bugzilla.mcs.anl.gov/globus/>

6. Technology Dependencies

This information provider depends on the following GT components:

- [Java WS Core](#)
- [GRAM4](#)

This information provider depends on the following 3rd party software:

- A working Perl installation
- A working Condor installation

7. Tested Platforms

Tested Platforms for this information provider:

- N/A

Tested containers for this information provider

- Java WS Core container

8. Backward Compatibility Summary

This information provider works with all GT4 and WS MDS releases.

9. Associated Standards

Associated standards for this Information Provider:

- N/A

10. For More Information

See [Chapter 2, GT 4.2.1: Hawkeye Information Provider Reference](#) for more information about this information provider.

Chapter 2. GT 4.2.1: Hawkeye Information Provider Reference

1. Overview

The Hawkeye Information Provider gathers Hawkeye data about Condor pool resources using the XML mapping of the GLUE schema and reports it to a GRAM4 service, which publishes it as resource properties. The official Condor site states the following:

Hawkeye utilizes the technologies already present in Condor and ClassAds to provide rich mechanisms for collecting, storing, and using information about computers. A Hawkeye system can be used to monitor various attributes of a collection of systems. The monitoring mechanism may also be used then to further the management of systems.

This information includes:

- basic host data (name, ID)
- processor information
- memory size
- OS name and version
- file system data
- processor load data
- other basic Condor host data

This information provider is included in the GT4 toolkit and is used for reporting GLUE Computing Element (CE) information. The GLUE resource property (as used by GRAM) collects information from two sources: the scheduler and the cluster information system (for example Ganglia or Hawkeye). These will be merged to form a single output resource property in the GLUE schema.

2. Prerequisites

1. A working Perl installation
2. A working Condor installation

3. Configuring

The following configuration is required for this information provider:

1. This section describes configuring for use in GRAM. The scheduler does not need to be configured, as GRAM already knows the scheduler bundle to use through other means. GT4.0 comes with providers which collect data from two cluster monitoring systems: * Ganglia * Hawkeye These are configured in

```
$GLOBUS_LOCATION/etc/globus_wsrp_mds_usefulrp/gluerp.xml
```

To collect data from Hawkeye, set the defaultProvider option in this file to:

```
<defaultProvider>shell /scratch/gt4b/I/libexec/globus-mds-cluster-hawkeye</defaultProvider>
```

If no cluster monitoring in this file is available, set the defaultProvider as follows: (this is the way the toolkit comes as installed)

```
<defaultProvider>none</defaultProvider>
```

4. Resource Properties

- The data gathered is published as part of the GLUECE RP

4.1. Namespace URI

- The GLUE namespace is: `http://mds.globus.org/glue/ce/1.1`
- The CE namespace is: `http://mds.globus.org/glue/ce/1.1`
- The XML namespace is: `http://www.w3.org/2001/XMLSchema`

5. Schema

- `$GLOBUS_LOCATION/share/schema/mds/usefulrp/ce.xsd`

6. Security Considerations

General security considerations associated with the container and all MDS services apply. See: [Aggregator Framework](#).

6.1. WS MDS Aggregator Services (Index Service and Trigger Service) Security Considerations

By default, the *aggregator sources* do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information. If a user or administrator changes that configuration so that a service's aggregator source uses credentials to acquire non-privileged data, then that user or administrator must configure the service's aggregator sink to limit access to authorized users.

7. Testing

N/A

8. Troubleshooting

Be sure that the `gluerp.xml` file is configured properly exactly as shown above. Simply having too much whitespace between the tags can cause errors.

Glossary

A

aggregator source

A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. WS MDS contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source.

Ganglia Information Provider

Ganglia Information Provider

Abstract

The Ganglia Information Provider collects information from two sources: the scheduler and the cluster information system (for example Ganglia or Hawkeye) using the XML mapping of the GLUE schema and reports it to a GRAM4 service, which publishes it as resource properties. The information from the scheduler and the cluster will be merged to form a single output resource property in the GLUE schema. The official Ganglia site states the following:

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters. It leverages widely used technologies such as XML for data representation, XDR for compact, portable data transport, and RRDtool for data storage and visualization

This provider publishes information such as the following:

- basic host data (name, ID)
- memory size
- OS name and version
- file system data
- processor load data
- other basic cluster data

You can download a PDF version of Ganglia Information Provider information [here](#)¹.

¹ ganglia.pdf

Table of Contents

1. Release Notes	1
1. Component Overview	1
2. Feature Summary	1
3. Changes Summary	1
4. Bug Fixes	1
5. Known Problems	1
6. Technology Dependencies	2
7. Tested Platforms	2
8. Backward Compatibility Summary	2
9. Associated Standards	2
10. For More Information	2
2. Reference Guide	3
1. Overview	3
2. Prerequisites	3
3. Configuring	3
4. Resource Properties	5
5. Schema	6
6. Security Considerations	6
7. Testing	6
8. Troubleshooting	6
Glossary	7

Chapter 1. GT 4.2.1 Release Notes: Ganglia Info Information Provider

1. Component Overview

The Ganglia Information Provider collects information from two sources: the scheduler and the cluster information system (for example Ganglia or Hawkeye) using the XML mapping of the GLUE schema and reports it to a GRAM4 service, which publishes it as resource properties. The information from the scheduler and the cluster will be merged to form a single output resource property in the GLUE schema. The official Ganglia site states the following:

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters. It leverages widely used technologies such as XML for data representation, XDR for compact, portable data transport, and RRDtool for data storage and visualization

This provider publishes information such as the following:

- basic host data (name, ID)
- memory size
- OS name and version
- file system data
- processor load data
- other basic cluster data

2. Feature Summary

Features new in release 4.2.1:

- This provider publishes information such as the following: basic host data (name, ID), memory size, OS name and version, file system data, processor load data, other basic cluster data.

3. Changes Summary

This is an existing information provider available from GT4 to GT 4.2.1.

4. Bug Fixes

There are ongoing fixed bugs for this information provider (see Bugzilla).

5. Known Problems

- There may currently be bugs for this information provider (see Bugzilla).

6. Technology Dependencies

This information provider depends on the following GT components:

- [Java WS Core](#)
- [GRAM4](#)

This information provider depends on the following 3rd party software:

- A working Perl installation
- A working Ganglia (gmond) installation

7. Tested Platforms

Tested Platforms for this information provider:

- N/A

Tested containers for this information provider

- Java WS Core container

8. Backward Compatibility Summary

This information provider works with all GT4 and WS MDS releases.

9. Associated Standards

Associated standards for this Information Provider:

- N/A

10. For More Information

See [Chapter 2, GT 4.2.1: Ganglia Information Provider Reference](#) for more information about this information provider.

Chapter 2. GT 4.2.1: Ganglia Information Provider Reference

1. Overview

The Ganglia Information Provider collects information from two sources: the scheduler and the cluster information system (for example Ganglia or Hawkeye) using the XML mapping of the GLUE schema and reports it to a GRAM4 service, which publishes it as resource properties. The information from the scheduler and the cluster will be merged to form a single output resource property in the GLUE schema. The official Ganglia site states the following:

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters. It leverages widely used technologies such as XML for data representation, XDR for compact, portable data transport, and RRDtool for data storage and visualization

This provider publishes information such as the following:

- basic host data (name, ID)
- memory size
- OS name and version
- file system data
- processor load data
- other basic cluster data

This information provider is included in the GT4 toolkit and is used for reporting GLUE Computing Element (CE) information. The GLUE resource property (as used by GRAM) collects information from two sources: the scheduler and the cluster information system (for example Ganglia or Hawkeye). These will be merged to form a single output resource property in the GLUE schema.

2. Prerequisites

1. A working Perl installation
2. A working Ganglia (gmond) installation

3. Configuring

The following configuration is required to use this information provider

1. **Make sure usefulRP is enabled (enabled by default in 4.2.1+):** Find the service's or resource's `server-config.wsdd` file, in this case `$GLOBUS_LOCATION/etc/globus_wsrp_mds_index/server-config.wsdd`, and make sure there is a `DefaultIndexService Handler` section similar to the following (if it is not there, you can copy and paste the following):

```

<service name="DefaultIndexService" provider="Handler" use="literal" style="document"
  <parameter name="providers"
    value="org.globus.mds.usefulrp.rpprovider.ResourcePropertyProviderCollection
    org.globus.wsrfl.impl.servicegroup.ServiceGroupRegistrationProvider
    GetRPPProvider
    GetMRPPProvider
    QueryRPPProvider
    DestroyProvider
    SetTerminationTimeProvider
    SubscribeProvider
    GetCurrentMessageProvider"/>
  <parameter name="rpProviderConfigFile"
    value="/etc/globus_wsrfl_mds_usefulrp/gluece-rpprovider-sample-config.xml"
  <parameter name="scope" value="Application"/>
  <parameter name="allowedMethods" value="*/>
  <parameter name="handlerClass" value="org.globus.axis.providers.RPCProvider"/>
  <parameter name="className" value="org.globus.mds.index.impl.DefaultIndexService"/>
  <wsdlFile>share/schema/mds/index/index_service.wsdl</wsdlFile>
</service>

```

Verify two things:

- In the *provider* section, one of the values should be `org.globus.mds.usefulrp.rpprovider.ResourcePropertyProviderCollection`
 - There should be an *rpProviderConfigFile* parameter with the OS-native file path to an RPPProvider config file. There should be a sample included by default. You can add to this config file or make your own config file and just make sure the path is updated here. This file contains all required information for generating one or more Resource Properties for the hosting service or resource.
2. **Enable the provider:** Edit the RPPProvider config file that was specified in the previous step to include the config block for Ganglia. The following configures the GLUE Resource Property provider with element producers using Ganglia to provide cluster information and PBS for scheduler information. This sample configures the provider to generate cluster information using Ganglia on the localhost with the default Ganglia port, and configures PBS as the scheduler information provider. The period of execution is set to 300 seconds for each element producer, but may be configured separately if desired. This configuration mirrors a common information provider setup in the GRAM4 ManagedJobExecutable service. You can also use a command line client to generate a new configuration file. For more information, see FIXME. Using the RPPProvider Framework, it is possible to generate this information in other services as well:

```

<nsl:ResourcePropertyProviderConfigArray
  xsi:type="nsl:ResourcePropertyProviderConfigArray"
  xmlns:nsl="http://mds.globus.org/rpprovider/2005/08"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <nsl:resourcePropertyProviderConfiguration xsi:type="nsl:resourcePropertyProvider
  <nsl:resourcePropertyName xsi:type="xsd:QName" xmlns:mds="http://mds.globus.org/g
  <nsl:resourcePropertyImpl xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovide
  <nsl:resourcePropertyElementProducers xsi:type="nsl:resourcePropertyElementProduc
  <nsl:className xsi:type="xsd:string">org.globus.mds.usefulrp.glue.GangliaElemen
  <nsl:arguments xsi:type="xsd:string">localhost</nsl:arguments>
  <nsl:arguments xsi:type="xsd:string">8649</nsl:arguments>

```

```

    <ns1:period xsi:type="xsd:int">300</ns1:period>
    <ns1:transformClass xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.tr
</ns1:resourcePropertyElementProducers>
<ns1:resourcePropertyElementProducers xsi:type="ns1:resourcePropertyElementProdu
    <ns1:className xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.produce
    <ns1:arguments xsi:type="xsd:string">libexec/globus-scheduler-provider-pbs</ns1
    <ns1:transformClass xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.tr
    <ns1:period xsi:type="xsd:int">300</ns1:period>
</ns1:resourcePropertyElementProducers>
</ns1:resourcePropertyProviderConfiguration>
</ns1:ResourcePropertyProviderConfigArray>

```

3. *Restart the container:* The information should now appear when querying the Index Service using [wsrf-query].

The configuration file format for the ResourcePropertyProviderCollection operation provider is simply the XML-serialized form of the ResourcePropertyProviderConfig stub object, as defined in the schema file rpprovider.xsd.

3.1. Generating an RPPProvider config file for GLUE

On resources running Ganglia:

1. Change working dir to \$GL/etc/gram-service-PBS (or -LSF, or -Condor, depending what you installed)
2. Run

```
mds-gluerp-configure pbs ganglia
```

If you're not using PBS, look for \$GLOBUS_LOCATION/globus-scheduler-provider-*. Use the value that appears there. You should see the following output:

```
Successfully wrote configuration output file to:
gluerp-config.xml
```

3. If Ganglia is running on the same (local) host as the container, and on the default port, then you can stop here. Otherwise, you will need to edit the gluerp-config.xml file to change the host and/or port. Open the file and look for the following lines:

```

<ns1:className xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xsi:type="xsd:string">org.globus.mds.usefulrp.glue.GangliaElementProducer
</ns1:className>
<ns1:arguments xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xsi:type="xsd:string">localhost</ns1:arguments>
<ns1:arguments xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xsi:type="xsd:string">8649</ns1:arguments>

```

The last two *ns1:arguments* lines are the host and port parameters, respectively. Change them to match your Ganglia installation host and port.

4. Resource Properties

- The data gathered is published as part of the GLUECE RP

4.1. Namespace URI

- The GLUE namespace is: `http://mds.globus.org/glue/ce/1.1`
- The CE namespace is: `http://mds.globus.org/glue/ce/1.1`
- The XML namespace is: `http://www.w3.org/2001/XMLSchema`

5. Schema

- `$GLOBUS_LOCATION/share/schema/mds/usefulrp/ce.xsd`

6. Security Considerations

General security considerations associated with the container and all MDS services apply. See: [Aggregator Framework](#).

6.1. WS MDS Aggregator Services (Index Service and Trigger Service) Security Considerations

By default, the *aggregator sources* do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information. If a user or administrator changes that configuration so that a service's aggregator source uses credentials to acquire non-privileged data, then that user or administrator must configure the service's aggregator sink to limit access to authorized users.

7. Testing

Once `gmond` is running and your configuration is complete, restart the container and the information should appear when querying the Index Service using `wsrf-query`.

8. Troubleshooting

Be sure that the `server-config.wsdd` and the `RPPProvider` configuration files are configured properly exactly as shown above. Simply having too much whitespace between the tags can cause errors.

Glossary

A

aggregator source

A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. WS MDS contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source.

Nagios Information Provider

Nagios Information Provider

Abstract

This information provider gathers data from a Nagios system and publishes it into the WS MDS in the standard GLUE schema. The Nagios WS MDS Information Provider gets its information directly from the Nagios v. 2.x `status.dat` file. (Support remains for the v 1.3 `status.log` file in the form of the alternate script `globus-mds-cluster-nagios1.3`) It includes some custom plugins that return the data in a format that the information provider can use. The official Nagios site states the following:

Nagios is a host and service monitor designed to inform you of network problems before your clients, end-users or managers do. It has been designed to run under the Linux operating system, but works fine under most *NIX variants as well. The monitoring daemon runs intermittent checks on hosts and services you specify using external plugins which return status information to Nagios. When problems are encountered, the daemon can send notifications out to administrative contacts in a variety of different ways (email, instant message, SMS, etc.). Current status information, historical logs, and reports can all be accessed via a web browser.

You can download a PDF version of Nagios Information Provider information [here](#)¹.

¹ nagios.pdf

Table of Contents

1. Release Notes	1
1. Component Overview	1
2. Feature Summary	1
3. Changes Summary	1
4. Bug Fixes	1
5. Known Problems	1
6. Technology Dependencies	1
7. Tested Platforms	2
8. Backward Compatibility Summary	2
9. Associated Standards	2
10. For More Information	2
2. Reference Guide	3
1. Overview	3
2. Prerequisites	3
3. Configuring	3
4. Resource Properties	8
5. Schema	8
6. Security Considerations	8
7. Testing	8
8. Troubleshooting	8
Glossary	10

Chapter 1. GT 4.2.1 Release Notes: Nagios Info Information Provider

1. Component Overview

This information provider gathers data from a Nagios system and publishes it into the WS MDS in the standard GLUE schema. The Nagios WS MDS Information Provider gets its information directly from the Nagios v. 2.x `status.dat` file. (Support remains for the v 1.3 `status.log` file in the form of the alternate script `globus-mds-cluster-nagios1.3`) It includes some custom plugins that return the data in a format that the information provider can use. The official Nagios site states the following:

Nagios is a host and service monitor designed to inform you of network problems before your clients, end-users or managers do. It has been designed to run under the Linux operating system, but works fine under most *NIX variants as well. The monitoring daemon runs intermittent checks on hosts and services you specify using external plugins which return status information to Nagios. When problems are encountered, the daemon can send notifications out to administrative contacts in a variety of different ways (email, instant message, SMS, etc.). Current status information, historical logs, and reports can all be accessed via a web browser.

2. Feature Summary

Features new in release 4.2.1:

- The Nagios WS MDS Information Provider gets its information directly from the Nagios v. 2.x `status.dat` file. (Support remains for the v 1.3 `status.log` file in the form of the alternate script `globus-mds-cluster-nagios1.3`) It includes some custom plugins that return the data in a format that the information provider can use.

3. Changes Summary

This is an existing information provider available from GT4.1.x to GT 4.2.1.

4. Bug Fixes

There are ongoing fixed bugs for this information provider (see Bugzilla).

5. Known Problems

- There may currently be bugs for this information provider (see Bugzilla).

6. Technology Dependencies

This information provider depends on the following GT components:

- [Java WS Core](#)
- [GRAM4](#)

This information provider depends on the following 3rd party software:

- A working Perl installation
- A working Condor installation

7. Tested Platforms

Tested Platforms for this information provider:

- N/A

Tested containers for this information provider

- Java WS Core container

8. Backward Compatibility Summary

This information provider works with all GT4 and WS MDS releases.

9. Associated Standards

Associated standards for this Information Provider:

- N/A

10. For More Information

See [Chapter 2, GT 4.2.1: Nagios Information Provider Reference](#) for more information about this information provider.

Chapter 2. GT 4.2.1: Nagios Information Provider Reference

1. Overview

This information provider gathers data from a Nagios system and publishes it into the WS MDS in the standard GLUE schema. The Nagios WS MDS Information Provider gets its information directly from the Nagios v. 2.x `status.dat` file. (Support remains for the v 1.3 `status.log` file in the form of the alternate script `globus-mds-cluster-nagios1.3`) It includes some custom plugins that return the data in a format that the information provider can use. The official Nagios site states the following:

Nagios is a host and service monitor designed to inform you of network problems before your clients, end-users or managers do. It has been designed to run under the Linux operating system, but works fine under most *NIX variants as well. The monitoring daemon runs intermittent checks on hosts and services you specify using external plugins which return status information to Nagios. When problems are encountered, the daemon can send notifications out to administrative contacts in a variety of different ways (email, instant message, SMS, etc.). Current status information, historical logs, and reports can all be accessed via a web browser.

2. Prerequisites

1. A working Perl installation
2. A working Nagios installation

3. Configuring

The following configuration is required for this information provider

1. Initial setup for Nagios WS MDS Information Provider:

The Nagios WS MDS Information Provider requires access to the status file of a Nagios installation that has been set up to provide the necessary information. Thus, if the WS MDS deployment is not on the same machine as the Nagios deployment, the Nagios' `status.dat` file must be exported in some manner to the machine with the WS MDS deployment. This could be accomplished via NFS or `rsync`, or a method of your choice.

The location of the `status.dat` file on the WS MDS deployment machine must be edited at the top of `$GLOBUS_LOCATION/libexec/globus-mds-cluster-nagios`.

2. Nagios Side Setup:

The Nagios' `services.cfg` file will need service stanzas for each host for each of the MDS plugins. The `service_description` fields by default are `MDS_OS`, `MDS_FS`, `MDS_PROC` and `MDS_MEM`. If there already exist tests that output the information needed, in the format needed, these values can be edited at the top of the `globus-mds-cluster-nagios` file. An example service stanza:

```
define service{
    use                               generic-service
```

```

host_name                tg-c001.uc.teragrid.org
service_description      check MDS_OS
is_volatile              0
check_period             24x7
max_check_attempts      3
normal_check_interval    3
retry_check_interval     5
contact_groups           teragrid-admins, nagios-mail
notification_interval    0
notification_period     24x7
notification_options     w,u,c,r
check_command            check_nrpe
}

```

There are several methods for Nagios to get information from remote hosts. NRPE, NSCA and SNMP are three of them. The Nagios WS MDS Information provider does not care how the information is obtained, as long as there is a correct service in services.cfg for each of the four types for each host.

There are sample plugins for the data discovery found in \$GLOBUS_LOCATION/share/nagios-plugins/check_os.mds, check_fs.mds, check_proc.mds and check_mem.mds. These are extremely simple shell script plugins--their important feature over other similar nagios plugins is that they output real information on stdout that is then used by the information provider.

3. The Sample Nagios MDS Information Provider Plugins:

The plugins included in the Nagios MDS Information provider package (check_os.mds, check_fs.mds, check_proc.mds, and check_mem.mds found in \$GLOBUS_LOCATION/share/nagios-plugins/). These, or functionally equivalent plugins will need to be installed on each host. As included in the Nagios WS MDS Information Provider package, these plugins expect a symlink to whatami to be present in their installed directory. They can be edited to point to the actual install of whatami.

The major difference between these plugins and other similar plugins that may already exist in your Nagios installation is that they provide specific information used by the Nagios WS MDS Information provider. This information is written to standard output when the plugin returns success. Thus, any plugin that returns the same information in the same format could be used instead.

Plugin Information:

MDS_OS

Takes no arguments, and prints the name of the operating system as determined by whatami -r

MDS_FS

Takes an argument of the mount point of the filesystem to be checked. Prints, in whitespace delimited columns, the name of the device associated with the mount point, the number of 1024-blocks on disk, the number of blocks used, the number of blocks available, the percent capacity, and the mount point. In other words, the output of df -Pk for that mount point.

MDS_PROC

Expects the number of CPUs for the host to be passed in as its first argument. If this number does not match the number found in

/proc/cpuinfo, the script will return with an error. If it does match, the script will print to standard out, the number of CPUs, the architecture (as determined by whatami -m) and the clock speed.

MDS_MEM

Takes no arguments, and outputs the total physical memory for your machine (as determined by free -m).

4. MDS Side Setup:

Unfortunately, the **[mds-gluerp-configure]** command does not produce a configuration file for a Nagios information provider. However, you can run **mds-gluerp-configure** as you would for Ganglia and simply edit the *resourcePropertyElementProducerConfig* section to include the correct ClassName and your actual OS-native file path to \$GLOBUS_LOCATION/libexec/globus-mds-cluster-nagios.

```
<ns1:resourcePropertyElementProducers xsi:type="ns1:resourcePropertyElementProducerConf
<ns1:className xsi:type="xsd:string" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
org.globus.mds.usefulrp.rpprovider.producers.ExternalProcessElementProducer</ns1:className
<ns1:arguments xsi:type="xsd:string" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
/PATH/TO/GLOBUS/LOCATION/HERE/libexec/globus-mds-cluster-nagios
</s1:arguments>
```

Ganglia can be configured as follows though it's only used here as a reference. Be sure to edit it as mentioned above for a working Nagios installation.

On resources running Ganglia:

1. Change working dir to \$GL/etc/gram-service-PBS (or -LSF, or -Condor, depending what you installed)
2. Run "mds-gluerp-configure pbs ganglia"

If you're not using PBS, look for \$GLOBUS_LOCATION/globus-scheduler-provider-*. Use the value that appears there. You should see the following output:

```
Successfully wrote configuration output file to:
gluerp-config.xml
```

3. If Ganglia is running on the same (local) host as the container, and on the default port, then you can stop here. Otherwise, you will need to edit the gluerp-config.xml file to change the host and/or port. Open the file and look for the following lines:

```
<ns1:className xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:type="xsd:string">org.globus.mds.usefulrp.glue.GangliaElementProducer
</ns1:className>
<ns1:arguments xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:type="xsd:string">localhost</ns1:arguments>
<ns1:arguments xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:type="xsd:string">8649</ns1:arguments>
```

The last two "ns1:arguments" lines are the host and port parameters, respectively. Change them to match your Ganglia installation host and port.

5. Configure cluster/sub-cluster provider:

Many of the Teragrid clusters are divided into sub-clusters based on some kind of characteristic, like processor speed, available memory, etc. This information will be aggregated and made visible to users on the basis of your site's cluster configuration file.

Copy `$GLOBUS_LOCATION/etc/globus_wsrp/mds_usefulrp/cluster-sample-config.xml` to `$GLOBUS_LOCATION/etc/globus_wsrp/mds_usefulrp/cluster_config.xml` and edit it.

This file is what organizes all of the site specific hosts returned into SubClusters. These SubClusters are groupings of hosts that share enough to be logically grouped in some manner. The groupings are arbitrary and you can have as many SubClusters as desired. Please consult the sample file to see an example of how each site needs to configure their cluster configurations.

6. Configure resource manager provider:

Please make the following edits to `$GL/etc/globus_wsrp/mds_usefulrp/scheduler-info.xml`:

1. Set `LRMSType` to "PBS-Torque", "PBS-OpenPBS", "PBS-PBSPro", "SGE", "Condor", "LoadLeveler", or "LSF"
2. Set `LRMSVersion` to the correct version
3. Set `HostName` to the URL of your GRAM service (like "https://tg-grid1.uc.teragrid.org:8443/wsrp/services/ManagedExecutableJobSer")
4. Set `GatekeeperPort` to the port running a GT4.0.1 gatekeeper [will be the alternate number until the service goes live]
5. Set `TotalCPUs` to the number of CPUs in your cluster

If you do this, you will need to add a transformation in the `SchedulerInfoElementProducer` section of the `gluerp-config.xml` file. That section is line 15-26, and the addition should be added after the `GLUESchedulerElementTransform` element. The transformation has the syntax:

```
<ns1:transformArguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/PATH/TO/GL/etc/globus_wsrp/mds_usefulrp/s
```

Here is an example of scheduler-info.xml files, though they contain fake version data - it's just to show what they look like syntactically:

```
<!-- NCSA example: useful for overriding the Info fields only -->
<ce:ComputingElementType xmlns:ce="http://mds.globus.org/glue/ce/1.1">
  <ce:Info ce:LRMSType="Fake-PBS" ce:LRMSVersion="FakeVersion"
    ce:GRAMVersion="4.0.1"
    ce:HostName="http://tg-login1.ncsa.teragrid.org:8888/wsrp/services/ManagedJobFac
    ce:GatekeeperPort="2119" ce:TotalCPUs="12800" />
</ce:ComputingElementType>
```

And here's an example gluerp-config.xml files that have the scheduler-info.xml transformation in the GLUES-chedulerElementTransform: * NCSA example

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- NCSA example: Configuration file for the MDS GLUECE resource property provider -->
<!-- -->

<ns1:ResourcePropertyProviderConfigArray
xmlns:ns1="http://mds.globus.org/rpprovider/2005/08"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ns1:ResourcePropertyProviderConfigArray">

  <ns1:resourcePropertyProviderConfiguration xsi:type="ns1:resourcePropertyProviderConf

    <ns1:resourcePropertyName
xmlns:mds="http://mds.globus.org/glue/ce/1.1"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="xsd:QName">mds:GLUECE</ns1:r

    <ns1:resourcePropertyImpl
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.GLUEResourceProperty</ns1:

    <ns1:resourcePropertyElementProducers
xsi:type="ns1:resourcePropertyElementProducerConfig">

      <ns1:className xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.producers.SchedulerInfoE

      <ns1:arguments xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:type="xsd:string">libexec/globus-scheduler-provider-pbs</ns1:arguments>

      <ns1:transformClass xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.transforms.GLUEScheduler

    <ns1:transformArguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/scratch/cbacon/INSTALL/etc/globus_wsrf_m

      <ns1:period xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="xsd:int">300</

  </ns1:resourcePropertyElementProducers>

  <ns1:resourcePropertyElementProducers xsi:type="ns1:resourcePropertyElementProducer

    <ns1:className xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.producers.URLElementProd

    <ns1:arguments xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:type="xsd:string">http://tg-clumon.ncsa.teragrid.org/glue.php</ns1:arguments>

    <ns1:transformClass xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.transforms.GLUECom
```

```
<ns1:transformArguments xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:type="xsd:string">./etc/globus_wsrp_mds_usefulrp/cluster_config.xml</ns1:tran
```

```
<ns1:transformArguments xsi:type="xsd:string" xmlns:xsd="http://www.w3.org/2001/XMLSche
  <ns1:period xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="xsd:int">300</
  </ns1:resourcePropertyElementProducers>
  </ns1:resourcePropertyProviderConfiguration>
</ns1:ResourcePropertyProviderConfigArray>
```

4. Resource Properties

- The data gathered is published as part of the GLUECE RP

4.1. Namespace URI

- The GLUE namespace is: <http://mds.globus.org/glue/ce/1.1>
- The CE namespace is: <http://mds.globus.org/glue/ce/1.1>
- The XML namespace is: <http://www.w3.org/2001/XMLSchema>

5. Schema

- `$(GLOBUS_LOCATION)/share/schema/mds/usefulrp/ce.xsd`

6. Security Considerations

General security considerations associated with the container and all MDS services apply. See: [Aggregator Framework](#).

6.1. WS MDS Aggregator Services (Index Service and Trigger Service) Security Considerations

By default, the *aggregator sources* do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information. If a user or administrator changes that configuration so that a service's aggregator source uses credentials to acquire non-privileged data, then that user or administrator must configure the service's aggregator sink to limit access to authorized users.

7. Testing

N/A

8. Troubleshooting

It should be noted that the `$(GLOBUS_LOCATION)/etc/globus_wsrp_mds_usefulrp/cluster_config.xml` file is **required** with the above configuration and, at a minimum, ALL host names **MUST** appear once (and only once) in the cluster-configuration file. If a host is retrieved from the Cluster information provider (whether it's CluMon, Ganglia, Nagios, etc) and it is not located in the cluster-configuration, an error will occur and the cluster information will not appear in the WS MDS Index. The error looks like:

```
2006-04-27 15:30:30,289 ERROR transforms.GLUEXMLOrganizer
[Thread-13,reorganizeGLUEDocument:826] Fatal error!
Unmapped hostname "lear.rcac.purdue.edu" in GLUE input data.
```

The `cfg:Name` in this file should correspond to the hostname as it will be reported by the monitoring system. For instance, if `clumon` refers to "tg-c001", the `cfg:UniqueID` will be "tg-c001". However, this Name should be unique across the Teragrid. For this purpose we can specify a string to append to the hostnames to get the unique names in the `gluERP-config.xml` file.

Here are two example `cluster-config.xml` files. The NCSA example is organized into subclusters based on various host machine characteristics. The Purdue example is not. I generated the `purdue` file by running a "for f in `seq -w 0 512`" loop to output the hostnames. They could be sorted into subclusters by an admin who knew how the machines were distributed.

http://software.teragrid.org/docs/ctss3/globus/4.0.1-r3/examples.mds/cluster_config.xml.ncsa

http://software.teragrid.org/docs/ctss3/globus/4.0.1-r3/examples.mds/cluster_config.xml.purdue

For your "Cluster" Name and UniqueID, please use the output of "tgwhereami|cut -f. -f1-2". For your "SubCluster"s please use the value for your Cluster with appended subcluster information.

For example, for Cluster Name, UniqueID "dtf.ncsa" SubCluster could be "dtf.ncsa-fastio", or "dtf.ncsa-<whatever>"

If you need to add a suffix to the hostnames, add the following entry below the `GLUEComputeElementTransform` element in the `gluERP-config.xml` file (around line 45 in the file):

```
<ns1:transformArguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">.ncsa.teragrid.org</ns1:transformArgument
```

Note also that if the `transformArgument` is used to modify host names (i.e. transforming "tg-c001" into "tg-c001.ncsa.teragrid.org"), it is *required* that the cluster configuration file contain the post-transformed hostnames rather than the original hostname. (i.e. "tg-c001.ncsa.teragrid.org")

Here are two example `gluERP-config.xml` files. The NCSA example has the suffix transformation because their `Clumon` install does not report the ".ncsa.teragrid.org" suffix on the hostnames. The Purdue example does not have a suffix transformation, because their `Ganglia` server reports on the FQDN of the host.

<http://software.teragrid.org/docs/ctss3/globus/4.0.1-r3/examples.mds/gluERP-config.xml.ncsa>

<http://software.teragrid.org/docs/ctss3/globus/4.0.1-r3/examples.mds/gluERP-config.xml.purdue>

Glossary

A

aggregator source

A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. WS MDS contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source.

GKrellm Information Provider

GKrellm Information Provider

Abstract

The GKrellm Information Provider monitors systems for statistics like memory usage, CPU load, network load and disk load. It does this by communicating with a GKrellmd server, which must be running on all of the machines which are to be monitored. This information provider will collect the data from all specified gkrellmd servers, expose the collected data as a WSRF resource, and publish the resource to the Index Service.

You can download a PDF version of GKrellm Information Provider information [here](#)¹.

¹ gkrellm.pdf

Table of Contents

1. Release Notes	1
1. Component Overview	1
2. Feature Summary	1
3. Changes Summary	1
4. Bug Fixes	1
5. Known Problems	1
6. Technology Dependencies	1
7. Tested Platforms	2
8. Backward Compatibility Summary	2
9. Associated Standards	2
10. For More Information	2
2. Reference Guide	3
1. Overview	3
2. Prerequisites	3
3. Configuring	3
4. Resource Properties	4
5. Schema	5
6. Security Considerations	5
7. Testing	6
8. Troubleshooting	6
Glossary	7

Chapter 1. GT 4.2.1 Release Notes: GKrellm Provider

1. Component Overview

The GKrellm Information Provider monitors systems for statistics like memory usage, CPU load, network load and disk load. It does this by communicating with a GKrellmd server, which must be running on all of the machines which are to be monitored. This information provider will collect the data from all specified gkrellmd servers, expose the collected data as a WSRF resource, and publish the resource to the Index Service.

2. Feature Summary

Features new in release 4.2.1:

- TODO: list any new features for this information provider for this release.

3. Changes Summary

This is a new component for 4.2.

4. Bug Fixes

No fixed bugs for this information provider.

5. Known Problems

The following problems and limitations are known to exist for WS MDS GKrellm Service at the time of the 4.2.1 release:

5.1. Limitations

- CPU load and current transfer rates are not being calculated correctly. They should be optional resource properties

5.2. Known Bugs

- There are currently no bugs for this information provider.

6. Technology Dependencies

The Index Service depends on the following GT components:

- [Java WS Core](#)

The Index Service depends on the following 3rd party software:

- All of the systems that the user wishes to observe via this information provider must be running gkrellmd version 2.2.7 or greater. More information on gkrellmd can be found at: <http://www.gkrellm.net/>

7. Tested Platforms

Tested Platforms for WS-MDS Index Service:

- Linux on i386
- Linux on x86_64

Tested containers for GKrellm Information Provider

- Java WS Core container

8. Backward Compatibility Summary

This is a new information provider with this version.

9. Associated Standards

Associated standards for WS GKrellm Information Provider:

- WS-ResourceProperties (WSRF-RP)
- WS-ResourceLifetime (WSRF-RL)
- WS-BaseFaults (WSRF-BF)
- WS-BaseNotification
- WS-Topics

10. For More Information

See [Chapter 2, GT 4.2.1: WS MDS GKrellm Information Provider Reference](#) for more information about this information provider.

Chapter 2. GT 4.2.1: WS MDS GKrellm Information Provider Reference

1. Overview

The GKrellm Information Provider monitors systems for statistics like memory usage, CPU load, network load and disk load. It does this by communicating with a GKrellmd server, which must be running on all of the machines which are to be monitored. This information provider will collect the data from all specified gkrellmd servers, expose the collected data as a WSRF resource, and publish the resource to the Index Service.

TBD: indicate which module this provider is implemented with (usefulrp or aggregator) and add link.

2. Prerequisites

TODO: list prerequisites

3. Configuring

3.1. Config file: gkrellmd.xml

There is one main configuration file for the gkrellm information service and that is: `$GLOBUS_LOCATION/etc/globus_wsrf_mds_gkrellm/gkrellmd.xml`

This is a very simple configuration file. To monitor a new system simply add a line to the `<providers>` section that points to the IP and port of the gkrellmd server that is to be monitored. For example if you wish to monitor a host with the IP 192.168.10.150 and there is a gkrellmd server running on the default port 19150 the `<providers>` would look like:

```
<providers>
<contacts>>192.168.10.150:19150</contact>
</providers>
```

When the container starts up, the GKrellm Information Provider will parse this file and start attempting to monitor all of the gkrellmd servers pointed to in the `<providers>` section. If it fails to connect to the server it will retry at regular intervals. To successfully acquire any information there must be a gkrellmd server running on the host.

All information will automatically be published in the DefaultIndexService running within the same container.

3.2. gkrellmd service

As stated above, a gkrellmd server must be running on in order to make this information provider useful. More information on installing and configuring gkrellmd can be found at: <http://www.gkrellm.net> .

A few basic details are described here. The default configuration file should be located at: `/etc/init.d/gkrellmd`. There is a section that describes which hosts are allowed to connect to the server and which are not. If you are having trouble getting information, make sure that this host where the container is running

is allowed access to the server. Additionally, verify that the max clients is high enough to sustain all of the hosts interested in connecting to the service:

```
max-clients 4

# List of hosts allowed to connect.  If no hosts are specified in a
# gkrellmd.conf file or on the command line, all hosts will be allowed.
#
allow-host localhost
allow-host 127.0.0.1
allow-host 192.168.10.*
```

If no allow-host lines are listed in the configuration file, the service will be available to all hosts.

This provider communicates with gkrellm via the protocol implemented in the gkrellm 2.2.7 protocol.

3.3. Example Web MDS

Since the resources presented by this service are automatically put into the DefaultIndexService the user can browse the information via Web MDS. Details on this can be found in [WebMDS](#).

In addition to this, we provide some simple webmds configuration files. These files can be found at: `$GLOBUS_LOCATION/etc/globus_wsrf_mds_gkrellm/gkrellm.xsl`.

1. Copy `$GLOBUS_LOCATION/etc/globus_wsrf_mds_gkrellm/gkrellm.xsl` to `$GLOBUS_LOCATION/lib/webmds/conf/gkrellm.xsl`
2. Copy `$GLOBUS_LOCATION/etc/globus_wsrf_mds_gkrellm/gkrellm.xsl` to `$GLOBUS_LOCATION/lib/webmds/xslfiles/gkrellm.xsl`
3. Then restart the tomcat container and the information should be available at: `http://<tomcat contact>/webmds/webmds?info=indexinfo&xsl=gkrellm.xsl`

4. Resource Properties

- Converts information provided by gkrellmd into resource properties which are then published into the Index Service.
 - Hostname
 - System description
 - Uptime
 - gkrellmd version
 - CPU Load information:
 - user space time
 - system space time
 - idle time

- nice time
- approximate load
- Memory Information:
 - Total system memory
 - Used memory
 - Available memory
 - Memory in buffered
 - Cached memory
 - Shared memory
 - Percentage Used
- Network Information:
 - Total number of bytes ever sent
 - Total number of bytes ever received
 - Number of bytes sent over a given interval
 - Number of bytes received over a given interval
- Disk Information:
 - Total number of bytes ever written
 - Total number of bytes ever read
 - Number of bytes written over a given interval
 - Number of bytes read over a given interval

4.1. Namespace URI

TODO: Include the Namespace URI, or provide a link to the Java API Documentation, which is required to construct the QName for the subscription to the Topic.

5. Schema

[gkrellm_port_type.wsdl](#)¹

6. Security Considerations

General security considerations associated with the container and all MDS services apply. See: [Aggregator Framework](#).

¹ http://viewcvs.globus.org/viewcvs.cgi/ws-mds/gkrellm/schema/mds/gkrellm/gkrellm_port_type.wsdl?rev=HEAD&only_with_tag=HEAD&content-type=text/vnd.viewcvs-markup

6.1. WS MDS Aggregator Services (Index Service and Trigger Service) Security Considerations

By default, the *aggregator sources* do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information. If a user or administrator changes that configuration so that a service's aggregator source uses credentials to acquire non-privileged data, then that user or administrator must configure the service's aggregator sink to limit access to authorized users.

7. Testing

TODO: add a simple test for this info provider

8. Troubleshooting

TODO: describe common issues users may experience with this info provider

Glossary

A

aggregator source

A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. WS MDS contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source.

Performance Benchmark Information Provider

Performance Benchmark Information Provider

Abstract

The basic idea of this Information Provider is to allow a set of performance benchmarks to be run and to aggregate the results of the test back into the WS MDS Index Service. To do this, each test must be configured separately, and each set of results will be placed in a separate Resource Property (RP). This simplifies the configuration and allows for easy client side querying of the results of an individual test. This information provider also includes a "Hello, World!" test that you should use alone to make sure that your environment is configured properly so that the other (more complicated) tests might have a chance of working as well.

You can download a PDF version of Performance Benchmark Information Provider information [here](#)¹.

¹ perf_benchmark.pdf

Table of Contents

1. Release Notes	1
1. Component Overview	1
2. Feature Summary	1
3. Changes Summary	1
4. Bug Fixes	1
5. Known Problems	1
6. Technology Dependencies	1
7. Tested Platforms	2
8. Backward Compatibility Summary	2
9. Associated Standards	2
10. For More Information	2
2. Reference Guide	3
1. Overview	3
2. Prerequisites	3
3. Configuring	4
4. Resource Properties	21
5. Schema	21
6. Security Considerations	21
7. Testing	21
8. Troubleshooting	21
Glossary	22

Chapter 1. GT 4.2.1 Release Notes: Performance Benchmark Information Provider

1. Component Overview

The basic idea of this Information Provider is to allow a set of performance benchmarks to be run and to aggregate the results of the test back into the WS MDS Index Service. To do this, each test must be configured separately, and each set of results will be placed in a separate Resource Property (RP). This simplifies the configuration and allows for easy client side querying of the results of an individual test. This information provider also includes a "Hello, World!" test that you should use alone to make sure that your environment is configured properly so that the other (more complicated) tests might have a chance of working as well.

2. Feature Summary

- [TODO: list new features of this info provider for this release]

3. Changes Summary

This is a new information provider for GT 4.2.

4. Bug Fixes

There are no fixed bugs for this information provider.

5. Known Problems

The following problems and limitations are known to exist for WS MDS GKrellm Service at the time of the 4.2.1 release:

5.1. Limitations

- [list]

5.2. Known Bugs

- There are currently no bugs for this information provider.

6. Technology Dependencies

The Index Service depends on the following GT components:

- [Java WS Core](#)

The Index Service depends on the following 3rd party software:

- All of the systems that the user wishes to observe via this information provider must be running gkrellmd version 2.2.7 or greater. More information on gkrellmd can be found at: <http://www.gkrellm.net/>

7. Tested Platforms

Tested Platforms for WS-MDS Index Service:

- Linux on i386
- Linux on x86_64

Tested containers for this Information Provider

- Java WS Core container

8. Backward Compatibility Summary

This is a new information provider with this version.

9. Associated Standards

Associated standards for this Information Provider:

- [list]

10. For More Information

See [Chapter 2, GT 4.2.1: Performance Benchmark Information Provider Reference](#) for more information about this information provider.

Chapter 2. GT 4.2.1: Performance Benchmark Information Provider Reference

1. Overview

The basic idea of this Information Provider is to allow a set of performance benchmarks to be run and to aggregate the results of the test back into the WS MDS Index Service. To do this, each test must be configured separately, and each set of results will be placed in a separate Resource Property (RP). This simplifies the configuration and allows for easy client side querying of the results of an individual test. This information provider also includes a "Hello, World!" test that you should use alone to make sure that your environment is configured properly so that the other (more complicated) tests might have a chance of working as well.

REQUIRED: indicate which module this provider is implemented with (usefulrp or aggregator) and add link.

2. Prerequisites

1. A cluster that has a shared file system between the login (or job submission) nodes and the backend (compute) nodes. For example, all backend (compute) nodes must be able to find a common file on some (arbitrary) mount point such as /nfs/shared/foo. As long as the program and output file can be run and written, respectively, there should be no trouble using this software.
2. A working GT4 installation with a configured WS-GRAM installation so that jobs can be submitted (via PBS, or Condor). A single running container is required for this information provider, and the job is submitted to this container.

To satisfy this requirement, you should be able to have a container running on the login node and be able to run a command such as the following on the command line without error:

```
$GLOBUS_LOCATION/bin/globusrun-ws -submit -Ft PBS -F \  
  https://MYHOST:MYPORT/wsrp/services/ManagedJobFactoryService \  
  -c /bin/true
```

The output should look similar to this:

```
-----  
Submitting job...Done.  
Job ID: uuid:7790abec-e5d9-11da-b93a-0014221d2259  
Termination time: 05/18/2006 19:15 GMT  
Current job state: Pending  
Current job state: Active  
Current job state: CleanUp  
Current job state: Done  
Destroying job...Done.  
-----
```

If this does not work without error on your system (where PBS should have been substituted for the scheduler installed on your system), please consult your system administrator or the WS-GRAM documentation located at: <http://www-unix.globus.org/toolkit/docs/4.0/execution/wsgam/user-index.html>

3. 3) A working and configured MPI installation on the backend nodes.
4. 4) An installation of the following programs that are accessible (i.e. can be run) from the backend (compute) nodes:

MPPTTEST: <http://www-unix.mcs.anl.gov/mpi/mpptest/>

MPIBENCH: <http://icl.cs.utk.edu/projects/llcbench/mpbench.html>

LLCBENCH: <http://icl.cs.utk.edu/projects/llcbench/index.html>

STREAM: <http://www.cs.virginia.edu/stream/>

3. Configuring

The following configuration is required for this information provider:

1. You must have the `$GLOBUS_LOCATION` and `$MPI_LOCATION` environment variables set. The `GLOBUS_LOCATION` must point to the root of a valid GT4 installation (such as `/nfs/software/globus-4.0.2`). Similarly, the `MPI_LOCATION` variable must point to the root of a valid MPI installation (such as `/nfs/software/mpich-1.2.7`). If you're unsure how to set these environment variables, check with your site administrator. Common ways to do this are like this:

```
[ bash users, try this ]
```

```
bash # export GLOBUS_LOCATION=/nfs/software/globus-4.0.2
bash # export MPI_LOCATION=/nfs/software/mpich-1.2.7
```

```
[ tcsh users, try this ]
```

```
tcsh $ setenv GLOBUS_LOCATION /nfs/software/globus-4.0.2
tcsh $ setenv MPI_LOCATION /nfs/software/mpich-1.2.7
```

2. The first major step is to make sure that your GT4 installation is a recent enough version and is compatible with this Information Provider. If you're running GT 4.0.2, you will need to update your installation with the May 5 WS-MDS RPPProvider patch located at:

<http://www.globus.org/toolkit/downloads/development/>

Install this patch by setting your `GLOBUS_LOCATION` to the appropriate directory and then running the following commands:

```
$GLOBUS_LOCATION/sbin/gpt-build -update gt4.0.2-wsmads-update-1.0-src_bundle.tar.gz
$GLOBUS_LOCATION/sbin/gpt-postinstall -force
```

3. When this step is complete, you should now have a file located at `$GLOBUS_LOCATION/etc/globus_wsrf_mds_usefulrp/gluece-rpprovider-sample-config.xml`. Copy this file to `$GLOBUS_LOCATION/etc/globus_wsrf_mds_usefulrp/rp-provider-config.xml`.
4. Edit your `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/server-config.wsdd` file.

Locate the `DefaultIndexService` section at the very bottom and replace it with the following:

```

<service name="DefaultIndexService" provider="Handler"
    use="literal" style="document">
    <parameter name="providers"
    value="org.globus.wsrfl.impl.servicegroup.ServiceGroupRegistrationProvider
    org.globus.mds.usefulrp.rpprovider.ResourcePropertyProviderCollection
    GetRPPProvider
    GetMRPPProvider
    QueryRPPProvider
    DestroyProvider
    SetTerminationTimeProvider
    SubscribeProvider
    GetCurrentMessageProvider"/>

    <parameter name="handlerClass"
    value="org.globus.axis.providers.RPCProvider"/>
    <parameter name="scope" value="Application"/>
    <parameter name="allowedMethods" value="*/>
    <parameter name="rpProviderConfigFile"
    value="/etc/globus_wsrfl_mds_usefulrp/rp-provider-config.xml"/>
    <parameter name="className"
    value="org.globus.mds.index.impl.DefaultIndexService"/>
    <wsdlFile>share/schema/mds/index/index_service.wsdl</wsdlFile>
</service>

```

- Download the [exec_wrapper script](#)¹ and the [perf_benchmark script](#)² to a location on your cluster that is accessible to the login node as well as the backend (compute) nodes. While the perf_benchmark is only run on the login node, the exec_wrapper must be accessible because it will be run on each of the backend nodes.

After placing these files in a suitable shared location, please make sure that they are executable by issuing the following shell command:

```
chmod a+x exec_wrapper perf_benchmark
```

- Finally, the rest of the configuration will be a matter of editing your `$GLOBUS_LOCATION/etc/globus_wsrfl_mds_usefulrp/rp-provider-config.xml` file. Assuming that the Cluster Monitoring and Scheduling data is already properly configured (which is outside the scope of this document), we need to make some edits.

3.1. Configuring the Information Provider to Run the HELLO WORLD performance test

The following block of XML is the configuration that is needed to enable the "Hello, World!" test in the Index Service. It uses the RPPProvider framework to create a Resource Property (RP) called PerfBM-HelloWorld that will contain the hello world information upon successful execution.



Note

Note that this configuration block must be added to the `$GLOBUS_LOCATION/etc/globus_wsrfl_mds_usefulrp/rp-provider-config.xml` file.

¹ exec_wrapper

² perf_benchmark

```

<ns1:resourcePropertyName xsi:type="xsd:QName"
  xmlns:perfbm="http://perfbm-testing">perfbm:PerfBM>HelloWorld</ns1:resourcePro
<ns1:resourcePropertyImpl
  xsi:type="xsd:string">org.globus.mds.usefulrp.rprovider.SingleValueResourcePr
<ns1:resourcePropertyElementProducers
  xsi:type="ns1:resourcePropertyElementProducerConfig">
<ns1:className xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">org.globus.mds.usefulrp.rprovide

<!-- *** SPECIFY THE SCRIPT TO RUN HERE *** -->

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/perf_benchmark<

<!-- *** BEGIN SCRIPT ARGUMENTS *** -->

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">SCHEDULER</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">https://MYHOST:MYPORT/wsrf/servic

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">4</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/tmp</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/exec_wrapper</n

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">hello_world</ns1:arguments>


<!-- *** END ARGUMENTS *** -->

<ns1:period xsi:type="xsd:int"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">60000</ns1:period>
</ns1:resourcePropertyElementProducers>

```

In the above, you must edit all of the argument lines.

1st argument	Specifies the absolute path location to the perf_benchmark script. It should be something like /nfs/home/user/perf_benchmark. Don't forget that this script must have executable permissions to run properly.
2nd argument	Specifies the scheduler to use. The only acceptable values here are Fork, PBS, and Condor. In theory the scheduler can be any scheduler that can be passed to WS-GRAM's globusrun-ws program.
3rd argument	Specifies the MJFS to which the job submission should be made. This must specify the MJFS to which the job submission should be made. This must be specifically a properly configured and running container. a properly configured and running container.

4th argument	Specifies the number of backend (compute) nodes that should be involved in the test that is to be run. For example, the value of 4 will run the job on 4 nodes and aggregate the results of the 4 hosts.
5th argument	Specify a valid temporary directory that the login node has file creation/removal and write access in. Some temporary files are used during the execution of this information provider and a directory must be specified where it can do this.
6th argument	Specifies the absolute path to the exec_wrapper program. It must be an absolute path and should be something like /nfs/home/user/exec_wrapper.  Important Do not forget that this script must have executable permissions to run properly.
7th argument	Specifies the test type that should be run. For this "Hello, World!" test, the value MUST be hello_world (as shown).

When this configuration block is placed properly within the resourcePropertyProviderConfiguration in the \$GLOBUS_LOCATION/etc/globus_wsrp_mds_usefulrp/rp-provider-config.xml file, the container can be restarted and when queried with a query such as this:

```
wsrf-query -s \
  https://MYHOST:MYPORT/wsrp/services/DefaultIndexService \
  "/*[local-name()='PerfBM-HelloWorld']"
```

You should see output that resembles the following:

```
<ns1:PerfBM-HelloWorld xmlns:ns1="http://perfbm-testing"
  xmlns:exw="http://perfbm.provider/2006/execWrapper"
  xmlns:pbo="http://perfbm.provider/2006/pbOutput">

  <pbo:perfBenchmarkOutputData>

    <exw:hostBenchmarkOutput>
      <exw:hostname>skynet-18</exw:hostname>
      <exw:programCommandLine>/bin/echo Hello, World!</exw:programCommandLine>
      <exw:startDate>Fri May 12 12:44:32 PDT 2006</exw:startDate>
      <exw:endDate>Fri May 12 12:44:32 PDT 2006</exw:endDate>
      <exw:testProgramOutput>
        Hello, World!
      </exw:testProgramOutput>
    </exw:hostBenchmarkOutput>

    <exw:hostBenchmarkOutput>
      <exw:hostname>skynet-19</exw:hostname>
      <exw:programCommandLine>/bin/echo Hello, World!</exw:programCommandLine>
      <exw:startDate>Fri May 12 12:44:32 PDT 2006</exw:startDate>
      <exw:endDate>Fri May 12 12:44:32 PDT 2006</exw:endDate>
      <exw:testProgramOutput>
        Hello, World!
      </exw:testProgramOutput>
    </exw:hostBenchmarkOutput>

    <exw:hostBenchmarkOutput>
```

```

<exw:hostname>skynet-94</exw:hostname>
<exw:programCommandLine>/bin/echo Hello, World!</exw:programCommandLine>
<exw:startDate>Fri May 12 12:44:32 PDT 2006</exw:startDate>
<exw:endDate>Fri May 12 12:44:32 PDT 2006</exw:endDate>
<exw:testProgramOutput>
Hello, World!
</exw:testProgramOutput>
</exw:hostBenchmarkOutput>

<exw:hostBenchmarkOutput>
<exw:hostname>skynet-93</exw:hostname>
<exw:programCommandLine>/bin/echo Hello, World!</exw:programCommandLine>
<exw:startDate>Fri May 12 12:44:32 PDT 2006</exw:startDate>
<exw:endDate>Fri May 12 12:44:32 PDT 2006</exw:endDate>
<exw:testProgramOutput>
Hello, World!
</exw:testProgramOutput>
</exw:hostBenchmarkOutput>

<pbo:perfBenchmarkErrors>
</pbo:perfBenchmarkErrors>

</pbo:perfBenchmarkOutputData>

</ns1:PerfBM-HelloWorld>

```

3.2. Configuring the Information Provider to Run the STREAM performance test

The following block of XML is the configuration that is needed to enable the Stream test in the Index Service. It uses the RPPProvider framework to create a Resource Property (RP) called PerfBM-Stream that will contain the stream output information upon successful execution.



Note

Note that this configuration block must be added to the `$GLOBUS_LOCATION/etc/globus_ws-rf_mds_usefulrp/rp-provider-config.xml` file.

```

<ns1:resourcePropertyName xsi:type="xsd:QName"
  xmlns:perfbm="http://perfbm-testing">perfbm:PerfBM-Stream</ns1:resourceProperty
  <ns1:resourcePropertyImpl
  xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.SingleValueResourcePr
  <ns1:resourcePropertyElementProducers
  xsi:type="ns1:resourcePropertyElementProducerConfig">
  <ns1:className xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">org.globus.mds.usefulrp.rpprovide

  <!-- *** SPECIFY THE SCRIPT TO RUN HERE *** -->

  <ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/perf_benchmark<

```

```

<!-- *** BEGIN SCRIPT ARGUMENTS *** -->

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">SCHEDULER</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">https://MYHOST:MYPORT/wsrf/service

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">4</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/tmp</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/exec_wrapper</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">stream</ns1:arguments>


<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/stream</ns1:arguments>


<!-- *** END ARGUMENTS *** -->

<ns1:period xsi:type="xsd:int"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">60000</ns1:period>
</ns1:resourcePropertyElementProducers>

```

In the above, you must edit all of the argument lines.

1st argument	<p>Specifies the absolute path location to the perf_benchmark script. It should be something like /nfs/home/user/perf_benchmark.</p> <p> Important</p> <p>Don't forget that this script must have executable permissions to run properly.</p>
2nd argument	<p>Specifies the scheduler to use. The only acceptable values here are Fork, PBS, and Condor. In theory the scheduler can be any scheduler that can be passed to WS-GRAM's globusrun-ws program.</p>
3rd argument	<p>Specifies the MJFS to which the job submission should be made. This must specify the MJFS to which the job submission should be made. This must be specify a properly configured and running container.</p>
4th argument	<p>Specifies the number of backend (compute) nodes that should be involved in the test that is to be run. For example, the value of 4 will run the job on 4 nodes and aggregate the results of the 4 hosts.</p>
5th argument	<p>Specify a valid temporary directory that the login node has file creation/removal and write access in. Some temporary files are used during the execution of this information provider and a directory must be specified where it can do this.</p>

6th argument	Specifies the absolute path to the <code>exec_wrapper</code> program. It must be an absolute path and should be something like <code>/nfs/home/user/exec_wrapper</code> .  Important Do not forget that this script must have executable permissions to run properly.
7th argument	Specifies the test type that should be run. For this Stream test, the value MUST be <code>stream</code> (as shown).
8th argument	Specifies the absolute path location to the stream binary. It should be something like <code>/nfs/home/user/stream</code> .

When this configuration block is placed properly within the `resourcePropertyProviderConfiguration` in the `$GLOBUS_LOCATION/etc/globus_wsrp/mds_usefulrp/rp-provider-config.xml` file, the container can be restarted and when queried with a query such as this:

```
wsrf-query -s https://MYHOST:MYPORT/wsrp/services/DefaultIndexService \
  "/*[local-name()='PerfBM-Stream']"
```

You should see output that resembles the following:

```
<ns1:PerfBM-Stream xmlns:ns1="http://perfbm-testing"
  xmlns:exw="http://perfbm.provider/2006/execWrapper"
  xmlns:pbo="http://perfbm.provider/2006/pbOutput">
  <pbo:perfBenchmarkOutputData>
  <exw:hostBenchmarkOutput>
  <exw:hostname>skynet-5</exw:hostname>

  <exw:programCommandLine>/nfs/home/mdsdev/neillm/stream/stream_d</exw:programCo

  <exw:startDate>Mon May 15 10:03:39 PDT 2006</exw:startDate>

  <exw:endDate>Mon May 15 10:04:13 PDT 2006</exw:endDate>

  <exw:testProgramOutput>
  -----
  This system uses 8 bytes per DOUBLE PRECISION word.
  -----
  Array size = 20005000, Offset = 0
  Total memory required = 457.9 MB.
  Each test is run 10 times, but only
  the *best* time for each is used.
  -----
  Your clock granularity/precision appears to be 10000 microseconds.
  Each test below will take on the order of 579999 microseconds.
  (= 57 clock ticks)
  Increase the size of the arrays if this shows that
  you are not getting at least 20 clock ticks per test.
  -----
  WARNING -- The above is only a rough guideline.
  For best results, please be sure you know the
  precision of your system timer.
```

```
-----  
Function      Rate (MB/s)  RMS time    Min time    Max time  
Copy:         444.5556    0.7240     0.7200     0.7300  
Scale:        450.8169    0.7180     0.7100     0.7200  
Add:          558.2791    0.8650     0.8600     0.8700  
Triad:        551.8621    0.8791     0.8700     0.9100  
</exw:testProgramOutput>
```

```
</exw:hostBenchmarkOutput>
```

```
<exw:hostBenchmarkOutput>
```

```
<exw:hostname>skynet-4</exw:hostname>
```

```
<exw:programCommandLine>/nfs/home/mdsdev/neillm/stream/stream_d</exw:programCo
```

```
<exw:startDate>Mon May 15 10:03:39 PDT 2006</exw:startDate>
```

```
<exw:endDate>Mon May 15 10:04:14 PDT 2006</exw:endDate>
```

```
<exw:testProgramOutput>
```

```
-----  
This system uses 8 bytes per DOUBLE PRECISION word.  
-----
```

```
Array size = 20005000, Offset = 0  
Total memory required = 457.9 MB.  
Each test is run 10 times, but only  
the *best* time for each is used.  
-----
```

```
Your clock granularity/precision appears to be 10000 microseconds.  
Each test below will take on the order of 589999 microseconds.  
(= 58 clock ticks)
```

```
Increase the size of the arrays if this shows that  
you are not getting at least 20 clock ticks per test.  
-----
```

```
WARNING -- The above is only a rough guideline.  
For best results, please be sure you know the  
precision of your system timer.  
-----
```

```
Function      Rate (MB/s)  RMS time    Min time    Max time  
Copy:         438.4658    0.7370     0.7300     0.7400  
Scale:        438.4658    0.7300     0.7300     0.7300  
Add:          551.8621    0.8710     0.8700     0.8800  
Triad:        545.5909    0.8810     0.8800     0.8900  
</exw:testProgramOutput>
```

```
</exw:hostBenchmarkOutput>
```

```
<exw:hostBenchmarkOutput>
```

```
<exw:hostname>skynet-2</exw:hostname>
```

```
<exw:programCommandLine>/nfs/home/mdsdev/neillm/stream/stream_d</exw:programCo
```

<exw:startDate>Mon May 15 10:03:39 PDT 2006</exw:startDate>

<exw:endDate>Mon May 15 10:04:14 PDT 2006</exw:endDate>

<exw:testProgramOutput>

This system uses 8 bytes per DOUBLE PRECISION word.

Array size = 20005000, Offset = 0
Total memory required = 457.9 MB.
Each test is run 10 times, but only
the *best* time for each is used.

Your clock granularity/precision appears to be 10000 microseconds.
Each test below will take on the order of 589999 microseconds.
(= 58 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.

WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.

Function	Rate (MB/s)	RMS time	Min time	Max time
Copy:	432.5405	0.7440	0.7400	0.7500
Scale:	438.4658	0.7360	0.7300	0.7400
Add:	545.5909	0.8860	0.8800	0.8900
Triad:	539.4607	0.8960	0.8900	0.9000

</exw:testProgramOutput>

</exw:hostBenchmarkOutput>

<exw:hostBenchmarkOutput>

<exw:hostname>skynet-3</exw:hostname>

<exw:programCommandLine>/nfs/home/mdsdev/neillm/stream/stream_d</exw:programCo

<exw:startDate>Mon May 15 10:03:39 PDT 2006</exw:startDate>

<exw:endDate>Mon May 15 10:04:14 PDT 2006</exw:endDate>

<exw:testProgramOutput>

This system uses 8 bytes per DOUBLE PRECISION word.

Array size = 20005000, Offset = 0
Total memory required = 457.9 MB.
Each test is run 10 times, but only
the *best* time for each is used.

Your clock granularity/precision appears to be 10000 microseconds.
Each test below will take on the order of 589999 microseconds.
(= 58 clock ticks)

Increase the size of the arrays if this shows that you are not getting at least 20 clock ticks per test.

 WARNING -- The above is only a rough guideline.
 For best results, please be sure you know the
 precision of your system timer.

Function	Rate (MB/s)	RMS time	Min time	Max time
Copy:	432.5405	0.7440	0.7400	0.7500
Scale:	438.4658	0.7360	0.7300	0.7400
Add:	545.5909	0.8870	0.8800	0.8900
Triad:	539.4607	0.8980	0.8900	0.9000

</exw:testProgramOutput>

</exw:hostBenchmarkOutput>

<pbo:perfBenchmarkErrors>
 </pbo:perfBenchmarkErrors>

</pbo:perfBenchmarkOutputData>
 </ns1:PerfBM-Stream>

3.3. Configuring the Information Provider to Run the MPPTTEST performance test

The following block of XML is the configuration that is needed to enable the MPPTest test in the Index Service. It uses the RPPProvider framework to create a Resource Property (RP) called PerfBM-MPPTest that will contain the mpttest output information upon successful execution.



Note

Note that this configuration block must be added to the \$GLOBUS_LOCATION/etc/globus_wsrp_mds_usefulrp/rp-provider-config.xml file.



Note

Also please note that this test cannot run without a properly set MPI_LOCATION environment variable.

```
<ns1:resourcePropertyName xsi:type="xsd:QName"
  xmlns:perfbm="http://perfbm-testing">perfbm:PerfBM-MPPTest</ns1:resourceProperty
  <ns1:resourcePropertyImpl
  xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.SingleValueResourcePr
  <ns1:resourcePropertyElementProducers
  xsi:type="ns1:resourcePropertyElementProducerConfig">
  <ns1:className xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">org.globus.mds.usefulrp.rpprovide

  <!-- *** SPECIFY THE SCRIPT TO RUN HERE *** -->

  <ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/perf_benchmark<
```

```

<!-- *** BEGIN SCRIPT ARGUMENTS *** -->

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">SCHEDULER</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">https://MYHOST:MYPORT/wsrf/service

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">4</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/tmp</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/exec_wrapper</n

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">mpptest</ns1:arguments>


<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/mpptest</ns1:ar


<!-- *** END ARGUMENTS *** -->

<ns1:period xsi:type="xsd:int"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">60000</ns1:period>
</ns1:resourcePropertyElementProducers>

```

In the above, you must edit all of the argument lines.

1st argument	<p>Specifies the absolute path location to the perf_benchmark script. It should be something like /nfs/home/user/perf_benchmark.</p> <p> Important</p> <p>Don't forget that this script must have executable permissions to run properly.</p>
2nd argument	<p>Specifies the scheduler to use. The only acceptable values here are Fork, PBS, and Condor. In theory the scheduler can be any scheduler that can be passed to WS-GRAM's globusrun-ws program.</p>
3rd argument	<p>Specifies the MJFS to which the job submission should be made. This must be specify a properly configured and running container..</p>
4th argument	<p>Specifies the number of backend (compute) nodes that should be involved in the test that is to be run. For example, the value of 4 will run the job on 4 nodes and aggregate the results of the 4 hosts.</p>
5th argument	<p>Specify a valid temporary directory that the login node has file creation/removal and write access in. Some temporary files are used during the execution of this information provider and a directory must be specified where it can do this.</p>

6th argument	<p>Specifies the absolute path to the <code>exec_wrapper</code> program. It must be an absolute path and should be something like <code>/nfs/home/user/exec_wrapper</code>.</p> <p> Important</p> <p>Do not forget that this script must have executable permissions to run properly.</p>
7th argument	<p>Specifies the test type that should be run. For this MPPTest test, the value MUST be <code>mpptest</code> (as shown).</p>
8th argument	<p>Specifies the absolute path location to the stream binary. It should be something like <code>/nfs/home/user/mpptest</code>.</p>

When this configuration block is placed properly within the `resourcePropertyProviderConfiguration` in the `$GLOBUS_LOCATION/etc/globus_wsrp_mds_usefulrp/rp-provider-config.xml` file, the container can be restarted and when queried with a query such as this:

```
wsrf-query -s https://MYHOST:MYPORT/wsrp/services/DefaultIndexService \
"//*[local-name()='PerfBM-MPPTest']"
```

You should see output that resembles the following:

```
<ns1:PerfBM-MPPTest xmlns:ns1="http://perfbm-testing"
  xmlns:exw="http://perfbm.provider/2006/execWrapper"
  xmlns:pbo="http://perfbm.provider/2006/pbOutput">
  <pbo:perfBenchmarkOutputData>
  <exw:hostBenchmarkOutput>
  <exw:hostname>skynet-2</exw:hostname>

  <exw:programCommandLine>/nfs/software/mpich/1.2.7/bin/mpirun -np 4
  /nfs/home/mdsdev/neillm/bin/mpptest</exw:programCommandLine>

  <exw:startDate>Wed May 17 11:52:35 PDT 2006</exw:startDate>

  <exw:endDate>Wed May 17 11:52:52 PDT 2006</exw:endDate>

  <exw:testProgramOutput>
  set default
  set font variable
  set curve window y 0.15 0.90
  set order d d d x y d
  title left 'time (us)', bottom 'Size (bytes)',
  top 'Comm Perf for MPI (skynet-2.isi.edu)',
  'type = blocking'

  #p0    p1    dist    len    ave time (us)    rate
  0      3      3       0      70.340000        0.00
  0      3      3      32      72.750000       439.863e+3
  0      3      3      64     106.750000       599.532e+3
  0      3      3      96     123.070000       780.044e+3
  0      3      3     128     124.070000       1.032e+6
  0      3      3     160     124.170000       1.289e+6
  0      3      3     192     124.270000       1.545e+6
```

0	3	3	224	124.300000	1.802e+6
0	3	3	256	124.490000	2.056e+6
0	3	3	288	124.510000	2.313e+6
0	3	3	320	124.450000	2.571e+6
0	3	3	352	124.750000	2.822e+6
0	3	3	384	124.810000	3.077e+6
0	3	3	416	125.150000	3.324e+6
0	3	3	448	125.390000	3.573e+6
0	3	3	480	126.970000	3.780e+6
0	3	3	512	128.340000	3.989e+6
0	3	3	544	133.480000	4.076e+6
0	3	3	576	136.000000	4.235e+6
0	3	3	608	137.880000	4.410e+6
0	3	3	640	163.280000	3.920e+6
0	3	3	672	185.420000	3.624e+6
0	3	3	704	186.560000	3.774e+6
0	3	3	736	186.730000	3.942e+6
0	3	3	768	186.740000	4.113e+6
0	3	3	800	186.690000	4.285e+6
0	3	3	832	186.820000	4.453e+6
0	3	3	864	186.900000	4.623e+6
0	3	3	896	187.000000	4.791e+6
0	3	3	928	187.110000	4.960e+6
0	3	3	960	187.200000	5.128e+6
0	3	3	992	187.250000	5.298e+6
0	3	3	1024	187.230000	5.469e+6

```
plot square
```

```
join
```

```
wait
```

```
new page
```

```
</exw:testProgramOutput>
```

```
</exw:hostBenchmarkOutput>
```

```
<exw:hostBenchmarkOutput>
```

```
<exw:hostname>skynet-3</exw:hostname>
```

```
<exw:programCommandLine>/nfs/software/mpich/1.2.7/bin/mpirun -np 4  
/nfs/home/mdsdev/neillm/bin/mpptest</exw:programCommandLine>
```

```
<exw:startDate>Wed May 17 11:52:35 PDT 2006</exw:startDate>
```

```
<exw:endDate>Wed May 17 11:52:52 PDT 2006</exw:endDate>
```

```
<exw:testProgramOutput>
```

```
set default
```

```
set font variable
```

```
set curve window y 0.15 0.90
```

```
set order d d d x y d
```

```
title left 'time (us)', bottom 'Size (bytes)',
```

```
top 'Comm Perf for MPI (skynet-3.isi.edu)',
```

```
'type = blocking'
```

#p0	p1	dist	len	ave time (us)	rate
0	3	3	0	94.780000	0.00
0	3	3	32	96.070000	333.090e+3
0	3	3	64	106.750000	599.532e+3
0	3	3	96	124.140000	773.320e+3
0	3	3	128	124.200000	1.031e+6
0	3	3	160	124.180000	1.288e+6
0	3	3	192	124.220000	1.546e+6
0	3	3	224	124.260000	1.803e+6
0	3	3	256	124.440000	2.057e+6
0	3	3	288	124.510000	2.313e+6
0	3	3	320	124.540000	2.569e+6
0	3	3	352	124.580000	2.825e+6
0	3	3	384	124.660000	3.080e+6
0	3	3	416	125.700000	3.309e+6
0	3	3	448	125.610000	3.567e+6
0	3	3	480	128.180000	3.745e+6
0	3	3	512	131.910000	3.881e+6
0	3	3	544	139.550000	3.898e+6
0	3	3	576	150.530000	3.826e+6
0	3	3	608	152.130000	3.997e+6
0	3	3	640	163.000000	3.926e+6
0	3	3	672	185.490000	3.623e+6
0	3	3	704	186.480000	3.775e+6
0	3	3	736	186.630000	3.944e+6
0	3	3	768	186.660000	4.114e+6
0	3	3	800	186.710000	4.285e+6
0	3	3	832	186.770000	4.455e+6
0	3	3	864	186.930000	4.622e+6
0	3	3	896	187.020000	4.791e+6
0	3	3	928	187.040000	4.962e+6
0	3	3	960	188.210000	5.101e+6
0	3	3	992	187.980000	5.277e+6
0	3	3	1024	187.180000	5.471e+6

plot square

join

wait

new page

</exw:testProgramOutput>

</exw:hostBenchmarkOutput>

<exw:hostBenchmarkOutput>

<exw:hostname>skynet-4</exw:hostname>

<exw:programCommandLine>/nfs/software/mpich/1.2.7/bin/mpirun -np 4
/nfs/home/mdsdev/neillm/bin/mpptest</exw:programCommandLine>

<exw:startDate>Wed May 17 11:52:35 PDT 2006</exw:startDate>

<exw:endDate>Wed May 17 11:52:52 PDT 2006</exw:endDate>

<exw:testProgramOutput>

```
set default
set font variable
set curve window y 0.15 0.90
set order d d d x y d
title left 'time (us)', bottom 'Size (bytes)',
top 'Comm Perf for MPI (skynet-4.isi.edu)',
'type = blocking'
```

#p0	p1	dist	len	ave time (us)	rate
0	3	3	0	77.630000	0.00
0	3	3	32	95.880000	333.751e+3
0	3	3	64	105.720000	605.373e+3
0	3	3	96	121.670000	789.019e+3
0	3	3	128	124.090000	1.032e+6
0	3	3	160	124.240000	1.288e+6
0	3	3	192	124.280000	1.545e+6
0	3	3	224	124.250000	1.803e+6
0	3	3	256	124.510000	2.056e+6
0	3	3	288	124.490000	2.313e+6
0	3	3	320	124.710000	2.566e+6
0	3	3	352	124.540000	2.826e+6
0	3	3	384	124.820000	3.076e+6
0	3	3	416	124.840000	3.332e+6
0	3	3	448	125.370000	3.573e+6
0	3	3	480	126.530000	3.794e+6
0	3	3	512	129.260000	3.961e+6
0	3	3	544	136.720000	3.979e+6
0	3	3	576	150.300000	3.832e+6
0	3	3	608	151.210000	4.021e+6
0	3	3	640	160.530000	3.987e+6
0	3	3	672	186.560000	3.602e+6
0	3	3	704	186.530000	3.774e+6
0	3	3	736	186.680000	3.943e+6
0	3	3	768	186.740000	4.113e+6
0	3	3	800	186.880000	4.281e+6
0	3	3	832	187.010000	4.449e+6
0	3	3	864	187.180000	4.616e+6
0	3	3	896	187.000000	4.791e+6
0	3	3	928	187.190000	4.958e+6
0	3	3	960	187.210000	5.128e+6
0	3	3	992	187.200000	5.299e+6
0	3	3	1024	187.200000	5.470e+6

```
plot square
```

```
join
```

```
wait
```

```
new page
```

```
</exw:testProgramOutput>
```

```
</exw:hostBenchmarkOutput>
```

```
<exw:hostBenchmarkOutput>
```

```
<exw:hostname>skynet-5</exw:hostname>
```

```
<exw:programCommandLine>/nfs/software/mpich/1.2.7/bin/mpirun -np 4  
/nfs/home/mdsdev/neillm/bin/mpptest</exw:programCommandLine>
```

```
<exw:startDate>Wed May 17 11:52:35 PDT 2006</exw:startDate>
```

```
<exw:endDate>Wed May 17 11:52:52 PDT 2006</exw:endDate>
```

```
<exw:testProgramOutput>
```

```
set default
```

```
set font variable
```

```
set curve window y 0.15 0.90
```

```
set order d d d x y d
```

```
title left 'time (us)', bottom 'Size (bytes)',
```

```
top 'Comm Perf for MPI (skynet-5.isi.edu)',
```

```
'type = blocking'
```

#p0	p1	dist	len	ave time (us)	rate
0	3	3	0	93.560000	0.00
0	3	3	32	94.490000	338.660e+3
0	3	3	64	106.770000	599.419e+3
0	3	3	96	124.160000	773.196e+3
0	3	3	128	123.250000	1.039e+6
0	3	3	160	124.260000	1.288e+6
0	3	3	192	124.250000	1.545e+6
0	3	3	224	124.310000	1.802e+6
0	3	3	256	124.410000	2.058e+6
0	3	3	288	124.470000	2.314e+6
0	3	3	320	124.520000	2.570e+6
0	3	3	352	124.730000	2.822e+6
0	3	3	384	125.580000	3.058e+6
0	3	3	416	125.630000	3.311e+6
0	3	3	448	125.200000	3.578e+6
0	3	3	480	127.930000	3.752e+6
0	3	3	512	127.280000	4.023e+6
0	3	3	544	129.450000	4.202e+6
0	3	3	576	143.250000	4.021e+6
0	3	3	608	155.510000	3.910e+6
0	3	3	640	164.380000	3.893e+6
0	3	3	672	185.450000	3.624e+6
0	3	3	704	186.550000	3.774e+6
0	3	3	736	186.610000	3.944e+6
0	3	3	768	186.680000	4.114e+6
0	3	3	800	186.780000	4.283e+6
0	3	3	832	186.770000	4.455e+6
0	3	3	864	186.910000	4.623e+6
0	3	3	896	186.990000	4.792e+6
0	3	3	928	187.000000	4.963e+6
0	3	3	960	187.100000	5.131e+6
0	3	3	992	187.420000	5.293e+6
0	3	3	1024	187.270000	5.468e+6

```
plot square
```

```
join
```

```
wait
```

```
new page
```

```
</exw:testProgramOutput>

</exw:hostBenchmarkOutput>

<pbo:perfBenchmarkErrors>
</pbo:perfBenchmarkErrors>

</pbo:perfBenchmarkOutputData>
</nsl:PerfBM-MPPTest>
```

3.3.1. Troubleshooting the MPPTTEST performance test

If you are seeing the following error after the container has been started, the most likely cause is that you have not properly set the `MPI_LOCATION` environment variable. Please set this to a suitable MPI location (such as `/nfs/software/mpich-1.2.7`) and restart the container.

```
2006-05-17 11:46:38,371 INFO impl.DefaultIndexService
[ServiceThread-12,processConfigFile:107] Reading default registration
configuration from file:
/scratch/mdsdev-neillm/gt4.0.2-plus-cvs/etc/globus_wsrp_mds_index/hierarch
2006-05-17 11:46:38,702 ERROR rprovider.ResourcePropertyProviderTask
[Thread-16,timerExpired:159] Unhandled exception during execution of
org.globus.mds.usefulrp.rprovider.producers.ExternalProcessElementProduce
java.lang.Exception: Exception while parsing child process stdout into
valid XML document: org.xml.sax.SAXException: Fatal Error: URI=null
Line=-1: Premature end of file.
```

3.4. Configuring the Information Provider to Run the MPPTTEST Logscale performance tests

The configuration for this test is exactly the same as the configuration for the [MPPTest](#) above, except for the RP name and 7th argument:

- Change the RP name from `PerfBM-MPPTest` to `PerfBM-MPPTest-LogScale`.
- Change the 7th argument of the `$GLOBUS_LOCATION/etc/globus_wsrp_mds_usefulrp/rp-provider-config.xml` file to be `mpptest-logscale` instead of `mpptest`.

3.5. Configuring the Information Provider to Run the MPPTTEST Bisect Logscale performance tests

The configuration for this test is exactly the same as the configuration for the [MPPTest](#) above, except for the RP name and 7th argument:

- Change the RP name from `PerfBM-MPPTest` to `PerfBM-MPPTest-BiSect-LogScale`.
- Change the 7th argument of the `$GLOBUS_LOCATION/etc/globus_wsrp_mds_usefulrp/rp-provider-config.xml` file to be `mpptest-bisect-logscale` instead of `mpptest`.

3.6. Configuring the Information Provider to Run the MPIBENCH/LLCBENCH performance tests

Information on this particular test is to be available at a future date.

4. Resource Properties

TODO: the resource properties the provider creates/collects/advertises/publishes

4.1. Namespace URI

TODO: Include the Namespace URI, or provide a link to the Java API Documentation, which is required to construct the QName for the subscription to the Topic.

5. Schema

TODO: link to schema files

6. Security Considerations

General security considerations associated with the container and all MDS services apply. See: [Aggregator Framework](#).

Additionally the user should consider that publishing system specific information as is the intention of this provider can have security ramifications.

6.1. WS MDS Aggregator Services (Index Service and Trigger Service) Security Considerations

By default, the *aggregator sources* do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information. If a user or administrator changes that configuration so that a service's aggregator source uses credentials to acquire non-privileged data, then that user or administrator must configure the service's aggregator sink to limit access to authorized users.

7. Testing

TODO: add a simple test for this info provider

8. Troubleshooting

TODO: describe common issues users may experience with this info provider

Glossary

A

aggregator source

A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. WS MDS contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source.

GridFTP Information Provider

GridFTP Information Provider

Abstract

This provider connects to a GridFTP server, reads its banner, and puts the banner in the Index Service. If the connection cannot be made in 30 seconds or some other error occurs, the server is marked as down. This provider is an execution aggregator provider. These providers are quite useful as is and additionally serve as an example of how to create others. The perl script and configuration file can be modified to the user's liking once installed. This provider has a perl script and an XML configuration file. The perl script is used to obtain information and formatted the information into xml output. The output is written to stdout where it is gathered and put into the Index Service. The XML file is of a common format that is explained [here](#). After installing the package the xml configuration files will be copied to `$GLOBUS_LOCATION/etc/globus_wsrft_mds_index/default_providers/` where the DefaultIndexService will find them and use them to register the provider.

You can download a PDF version of GridFTP Information Provider information [here](#)¹.

¹ [gridftp_info.pdf](#)

Table of Contents

1. Release Notes	1
1. Component Overview	1
2. Feature Summary	1
3. Changes Summary	1
4. Bug Fixes	1
5. Known Problems	1
6. Technology Dependencies	1
7. Tested Platforms	1
8. Backward Compatibility Summary	2
9. Associated Standards	2
10. For More Information	2
2. Reference Guide	3
1. Overview	3
2. Prerequisites	3
3. Configuring	3
4. Resource Properties	4
5. Schema	4
6. Security Considerations	4
7. Testing	4
8. Troubleshooting	5
Glossary	6

Chapter 1. GT 4.2.1 Release Notes: GridFTP Info Information Provider

1. Component Overview

This provider connects to a GridFTP server, reads its banner, and puts the banner in the Index Service. If the connection cannot be made in 30 seconds or some other error occurs, the server is marked as down. This provider is an execution aggregator provider. These providers are quite useful as is and additionally serve as an example of how to create others. The perl script and configuration file can be modified to the user's liking once installed. This provider has a perl script and an XML configuration file. The perl script is used to obtain information and formatted the information into xml output. The output is written to stdout where it is gathered and put into the Index Service. The XML file is of a common format that is explained [here](#). After installing the package the xml configuration files will be copied to `$GLOBUS_LOCATION/etc/globus_wsrft_mds_index/default_providers/` where the DefaultIndexService will find them and use them to register the provider.

2. Feature Summary

Features new in release 4.2.1:

- This provider determines if a GridFTP server is online.

3. Changes Summary

This is a new information provider for GT 4.2.1.

4. Bug Fixes

There are no fixed bugs for this information provider.

5. Known Problems

- There are currently no bugs for this information provider.

6. Technology Dependencies

This information provider depends on the following GT components:

- [Java WS Core](#)

This information provider depends on the following 3rd party software:

- A working Perl installation

7. Tested Platforms

Tested Platforms for this information provider:

- N/A

Tested containers for this information provider

- Java WS Core container

8. Backward Compatibility Summary

This is a new information provider with this version.

9. Associated Standards

Associated standards for this Information Provider:

- N/A

10. For More Information

See [Chapter 2, GT 4.2.1: GridFTP Information Provider Reference](#) for more information about this information provider.

Chapter 2. GT 4.2.1: GridFTP Information Provider Reference

1. Overview

This provider connects to a GridFTP server, reads its banner, and puts the banner in the Index Service. If the connection cannot be made in 30 seconds or some other error occurs, the server is marked as down. This provider is an execution aggregator provider. These providers are quite useful as is and additionally serve as an example of how to create others. The perl script and configuration file can be modified to the user's liking once installed. This provider has a perl script and an XML configuration file. The perl script is used to obtain information and formatted the information into xml output. The output is written to stdout where it is gathered and put into the Index Service. The XML file is of a common format that is explained [here](#). After installing the package the xml configuration files will be copied to `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/default_providers/` where the DefaultIndexService will find them and use them to register the provider.

This is an execution aggregator source based information provider.

2. Prerequisites

1. A working Perl installation

3. Configuring

The following configuration is required for this information provider:

1. In order to grant permission to the installed perl scripts the user must edit the file: `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/jndi-config.xml` such that the tags:

```
<parameter>
  <name>executableMappings</name>
  <value>
    aggr-test=aggregator-exec-test.sh, pingexec=example-ping-exec
  </value>
</parameter>
```

includes the new provider. For example:

```
<parameter>
  <name>executableMappings</name>
  <value>
    gridftp-info=gridftp-banner.pl
  </value>
</parameter>
```

2. This provider connects to a gridftp server, reads its banner, and puts the banner in the index. If the connection cannot be made in 30 seconds, or some other error occurs the server is marked as down. To change what server is to be monitored edit the file:

GLOBUS_LOCATION/etc/globus_wsrif_mds_index/default_providers/gridftp-info-reg.xml

and change the line:

```
<RegistrantEPR
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"
  xmlns:agg="http://mds.globus.org/aggregator/types">
  <wsa:Address>ftp://www.SITE.COM:2811</wsa:Address>
</RegistrantEPR>
```

from "ftp://www.SITE.COM:2811" to the server of your choice.

4. Resource Properties

- Data appears as part of the ServiceGroupEntry RP

4.1. Namespace URI

TODO: Include the Namespace URI, or provide a link to the Java API Documentation, which is required to construct the QName for the subscription to the Topic.

5. Schema

- N/A

6. Security Considerations

General security considerations associated with the container and all MDS services apply. See: [Aggregator Framework](#).

6.1. WS MDS Aggregator Services (Index Service and Trigger Service) Security Considerations

By default, the *aggregator sources* do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information. If a user or administrator changes that configuration so that a service's aggregator source uses credentials to acquire non-privileged data, then that user or administrator must configure the service's aggregator sink to limit access to authorized users.

7. Testing

N/A

8. Troubleshooting

N/A

Glossary

A

aggregator source

A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. WS MDS contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source.

Web Check Information Provider

Web Check Information Provider

Abstract

This provider connects to a web server. If the connection can successful be made the server is registered as alive, otherwise it is down. This provider is an execution aggregator provider. These providers are quite useful as is and additionally serve as an example of how to create others. The perl script and configuration file can be modified to the users liking once installed. This provider has a perl script and an xml configuration file. The perl script is used to obtain information and formatted the information into xml output. The output is written to stdout where it is gathered and put into the Index Service. The xml file is of a common format that is explained [here](#). After installing the package the xml configuration files will be copied to `$GLOBUS_LOCATION/etc/globus_wsrp_mds_index/default_providers/` where the DefaultIndexService will find them and use them to register the provider.

You can download a PDF version of Web Check Information Provider information [here](#)¹.

¹ web_info.pdf

Table of Contents

1. Release Notes	1
1. Component Overview	1
2. Feature Summary	1
3. Changes Summary	1
4. Bug Fixes	1
5. Known Problems	1
6. Technology Dependencies	1
7. Tested Platforms	1
8. Backward Compatibility Summary	2
9. Associated Standards	2
10. For More Information	2
2. Reference Guide	3
1. Overview	3
2. Prerequisites	3
3. Configuring	3
4. Resource Properties	4
5. Schema	4
6. Security Considerations	4
7. Testing	4
8. Troubleshooting	5
Glossary	6

Chapter 1. GT 4.2.1 Release Notes: Web Check Information Provider

1. Component Overview

This provider connects to a web server. If the connection can successful be made the server is registered as alive, otherwise it is down. This provider is an execution aggregator provider. These providers are quite useful as is and additionally serve as an example of how to create others. The perl script and configuration file can be modified to the users liking once installed. This provider has a perl script and an xml configuration file. The perl script is used to obtain information and formatted the information into xml output. The output is written to stdout where it is gathered and put into the Index Service. The xml file is of a common format that is explained [here](#). After installing the package the xml configuration files will be copied to `$GLOBUS_LOCATION/etc/globus_wsrp_mds_index/default_providers/` where the DefaultIndexService will find them and use them to register the provider.

2. Feature Summary

Features new in release 4.2.1:

- This provider determines if a Web (HTTP) server is online.

3. Changes Summary

This is a new information provider for GT 4.2.1.

4. Bug Fixes

There are no fixed bugs for this information provider.

5. Known Problems

- There are currently no bugs for this information provider.

6. Technology Dependencies

This information provider depends on the following GT components:

- [Java WS Core](#)

This information provider depends on the following 3rd party software:

- A working Perl installation

7. Tested Platforms

Tested Platforms for this information provider:

- N/A

Tested containers for this information provider

- Java WS Core container

8. Backward Compatibility Summary

This is a new information provider with this version.

9. Associated Standards

Associated standards for this Information Provider:

- N/A

10. For More Information

See [Chapter 2, GT 4.2.1: Web Check Information Provider Reference](#) for more information about this information provider.

Chapter 2. GT 4.2.1: Web Check Information Provider Reference

1. Overview

This provider connects to a web server. If the connection can successful be made the server is registered as alive, otherwise it is down. This provider is an execution aggregator provider. These providers are quite useful as is and additionally serve as an example of how to create others. The perl script and configuration file can be modified to the users liking once installed. This provider has a perl script and an xml configuration file. The perl script is used to obtain information and formatted the information into xml output. The output is written to stdout where it is gathered and put into the Index Service. The xml file is of a common format that is explained [here](#). After installing the package the xml configuration files will be copied to `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/default_providers/` where the DefaultIndexService will find them and use them to register the provider.

This is an execution aggregator source based information provider.

2. Prerequisites

1. A working Perl installation

3. Configuring

The following configuration is required for this information provider:

1. In order to grant permission to the installed perl scripts the user must edit the file: `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/jndi-config.xml` such that the tags:

```
<parameter>
  <name>executableMappings</name>
  <value>
    aggr-test=aggregator-exec-test.sh, pingexec=example-ping-exec
  </value>
</parameter>
```

includes the new provider. For example:

```
<parameter>
  <name>executableMappings</name>
  <value>
    web-info=web-check.pl
  </value>
</parameter>
```

2. This provider connects to a web server. If the connection can successful be made the server is registered as alive, otherwise it is down. To change what server is monitored edit the tag:

`$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/default_providers/web-info-reg.xml`

and change the line:

```
<RegistrantEPR
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"
  xmlns:agg="http://mds.globus.org/aggregator/types">
  <wsa:Address>http://www.SITE.COM</wsa:Address>
</RegistrantEPR>
```

from "http://www.SITE.COM" to the webserver of your choice.

4. Resource Properties

- Data appears as part of the ServiceGroupEntry RP

4.1. Namespace URI

TODO: Include the Namespace URI, or provide a link to the Java API Documentation, which is required to construct the QName for the subscription to the Topic.

5. Schema

- N/A

6. Security Considerations

General security considerations associated with the container and all MDS services apply. See: [Aggregator Framework](#).

Additionally the user should consider that publishing system specific information as is the intention of this provider can have security ramifications.

6.1. WS MDS Aggregator Services (Index Service and Trigger Service) Security Considerations

By default, the *aggregator sources* do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information. If a user or administrator changes that configuration so that a service's aggregator source uses credentials to acquire non-privileged data, then that user or administrator must configure the service's aggregator sink to limit access to authorized users.

7. Testing

N/A

8. Troubleshooting

N/A

Glossary

A

aggregator source

A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. WS MDS contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source.

Cert Check Information Provider

Cert Check Information Provider

Abstract

This provider will monitor the life of a remote certificate. It connects to an SSL TCP listener and obtains the certificate from it. It then checks the certificate for subject, start date, end data, and email address. All of this information is put in the Index Service if it is successfully obtained. This provider is an execution aggregator provider. These providers are quite useful as is and additionally serve as an example of how to create others. The perl script and configuration file can be modified to the user's liking once installed. This provider has a perl script and an XML configuration file. The perl script is used to obtain information and format the information into XML output. The output is written to stdout, where it is gathered and put into the Index Service. The XML file is of a common format that is explained [here](#). After installing the package, the XML configuration files will be copied to `$GLOBUS_LOCATION/etc/globus_ws-rf_mds_index/default_providers/` where the `DefaultIndexService` will find them and use them to register the provider.

You can download a PDF version of Cert Check Information Provider information [here](#)¹.

¹ cert_check.pdf

Table of Contents

1. Release Notes	1
1. Component Overview	1
2. Feature Summary	1
3. Changes Summary	1
4. Bug Fixes	1
5. Known Problems	1
6. Technology Dependencies	1
7. Tested Platforms	2
8. Backward Compatibility Summary	2
9. Associated Standards	2
10. For More Information	2
2. Reference Guide	3
1. Overview	3
2. Prerequisites	3
3. Configuring	3
4. Resource Properties	4
5. Schema	4
6. Security Considerations	4
7. Testing	4
8. Troubleshooting	5
Glossary	6

Chapter 1. GT 4.2.1 Release Notes: Cert Check Info Information Provider

1. Component Overview

This provider will monitor the life of a remote certificate. It connects to an SSL TCP listener and obtains the certificate from it. It then checks the certificate for subject, start date, end data, and email address. All of this information is put in the Index Service if it is successfully obtained. This provider is an execution aggregator provider. These providers are quite useful as is and additionally serve as an example of how to create others. The perl script and configuration file can be modified to the user's liking once installed. This provider has a perl script and an XML configuration file. The perl script is used to obtain information and format the information into XML output. The output is written to stdout, where it is gathered and put into the Index Service. The XML file is of a common format that is explained [here](#). After installing the package, the XML configuration files will be copied to `$GLOBUS_LOCATION/etc/globus_ws-rf_mds_index/default_providers/` where the `DefaultIndexService` will find them and use them to register the provider.

2. Feature Summary

Features new in release 4.2.1:

- This provider monitors the life of a remote certificate.

3. Changes Summary

This is a new information provider for GT 4.2.1.

4. Bug Fixes

There are no fixed bugs for this information provider.

5. Known Problems

- There are currently no bugs for this information provider.

6. Technology Dependencies

This information provider depends on the following GT components:

- [Java WS Core](#)

This information provider depends on the following 3rd party software:

- A working Perl installation

7. Tested Platforms

Tested Platforms for this information provider:

- N/A

Tested containers for this information provider

- Java WS Core container

8. Backward Compatibility Summary

This is a new information provider with this version.

9. Associated Standards

Associated standards for this Information Provider:

- N/A

10. For More Information

See [Chapter 2, GT 4.2.1: Cert Check Information Provider Reference](#) for more information about this information provider.

Chapter 2. GT 4.2.1: Cert Check Information Provider Reference

1. Overview

This provider will monitor the life of a remote certificate. It connects to an SSL TCP listener and obtains the certificate from it. It then checks the certificate for subject, start date, end data, and email address. All of this information is put in the Index Service if it is successfully obtained. This provider is an execution aggregator provider. These providers are quite useful as is and additionally serve as an example of how to create others. The perl script and configuration file can be modified to the user's liking once installed. This provider has a perl script and an XML configuration file. The perl script is used to obtain information and format the information into XML output. The output is written to stdout, where it is gathered and put into the Index Service. The XML file is of a common format that is explained [here](#). After installing the package, the XML configuration files will be copied to `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/default_providers/` where the `DefaultIndexService` will find them and use them to register the provider.

This is an execution aggregator source-based information provider.

2. Prerequisites

1. A working Perl installation

3. Configuring

The following configuration is required for this information provider:

1. In order to grant permission to the installed perl scripts the user must edit the file: `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/jndi-config.xml` such that the tags:

```
<parameter>
  <name>executableMappings</name>
  <value>
    aggr-test=aggregator-exec-test.sh, pingexec=example-ping-exec
  </value>
</parameter>
```

includes the new provider. For example:

```
<parameter>
  <name>executableMappings</name>
  <value>
    cert-info=cert-check.pl
  </value>
</parameter>
```

2. This provider will monitor the life of a remote certificate. It connects to an ssl TCP listener and obtains the certificate from it. It then checks the cert for subject, start date, end data, and email address. All of this information is put in the index if it is successfully obtained. To change what server is monitored edit the file:

GLOBUS_LOCATION/etc/globus_wsrf_mds_index/default_providers/cert-info-reg.xml

and change the line:

```
<RegistrantEPR
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"
  xmlns:agg="http://mds.globus.org/aggregator/types">
  <wsa:Address>https://www.SITE.COM:8443</wsa:Address>
</RegistrantEPR>
```

from "https://www.SITE.COM:8443" to the ssl server of your choice.

4. Resource Properties

- Data appears as part of the ServiceGroupEntry RP

4.1. Namespace URI

TODO: Include the Namespace URI, or provide a link to the Java API Documentation, which is required to construct the QName for the subscription to the Topic.

5. Schema

- N/A

6. Security Considerations

General security considerations associated with the container and all MDS services apply. See: [Aggregator Framework](#).

6.1. WS MDS Aggregator Services (Index Service and Trigger Service) Security Considerations

By default, the *aggregator sources* do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information. If a user or administrator changes that configuration so that a service's aggregator source uses credentials to acquire non-privileged data, then that user or administrator must configure the service's aggregator sink to limit access to authorized users.

7. Testing

N/A

8. Troubleshooting

N/A

Glossary

A

aggregator source

A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. WS MDS contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source.