

GT 4.2.1 Index Service: Public Interface Guide

GT 4.2.1 Index Service: Public Interface Guide

Table of Contents

1. APIs	1
1. Programming Model Overview	1
2. WS and WSDL	2
1. Protocol overview	2
2. Operations	2
3. WS MDS Aggregator Framework Resource Properties	3
4. Faults	3
5. WSDL and Schema Definition	3
3. Configuring the WS MDS Index Service	4
1. Configuration overview	4
2. Defining the Aggregator Sources	4
4. Graphical User Interface	6
5. Configuring an executable to retrieve information	7
1. Interface introduction	7
2. Syntax of the interface	7
A. Errors	8
Glossary	9

List of Tables

A.1. WS MDS Index Service Error Messages	8
--	---

Chapter 1. APIs

1. Programming Model Overview

Index Service queries are performed using resource property requests; consult [Java WS Core](#) for details.

The contents of an index are maintained using the aggregator framework programming model, and can receive data from any *aggregator source*. Information about how to configure existing aggregator sources (such as the aggregator sources distributed with the Globus Toolkit, which include one that polls for resource property information, one that collects resource property information through subscription/notification, and one that collects information by executing an executable program) is found in the [Aggregator Sources Reference](#); information about how to create new aggregator sources can be found in [Developer's Guide](#).

Chapter 2. WS and WSDL

1. Protocol overview

The Aggregator Framework builds on the [WS-ServiceGroup](#)¹ and [WS-ResourceLifetime](#)² specifications. Those specifications should be consulted for details on the syntax of each operation.

Each Aggregator Framework is represented as a WS-ServiceGroup (specifically, an AggregatorServiceGroup).

Resources may be registered to an AggregatorServiceGroup using the AggregatorServiceGroup Add operation. Each registration will be represented as a ServiceGroupEntry resource. Resources may be *registered* to an AggregatorServiceGroup using the service group add operation, which will cause an entry to be added to the service group.

The entry will include configuration parameters for the *aggregator source*; when the registration is made, the following will happen:

1. The appropriate aggregation source and sinks will be informed,
2. the aggregator source will begin collecting data and inserting it into the corresponding service group entry,
3. and the aggregator sink will begin processing the information in the service group entries.

The method of collection by source and processing by the sink is dependent on the particular instantiation of the aggregator framework (see [per-source documentation](#) for source information and [per-service documentation](#) for sink information - for example the [Index Service](#) and the [Trigger Service](#).)

2. Operations

2.1. AggregatorServiceGroup

- `add`: This operation is used to register a specified resource with the Aggregator Framework. In addition to the requirements made by the WS-ServiceGroup specification, the Content element of each registration must be an AggregatorContent type, with the AggregatorConfig element containing configuration information specific to each source and sink (documented in the [System Administrator's Guide](#)).

2.2. AggregatorServiceGroupEntry

- `setTerminationTime`: This operation can be used to set the termination time of the registration, as detailed in WS-ResourceLifetime.

¹ http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/servicegroup/sgw-2.wsdl?revision=1.2&view=markup&pathrev=globus_4_2_branch

² http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/lifetime/rlw-2.wsdl?revision=1.2&view=markup&pathrev=globus_4_2_branch

3. WS MDS Aggregator Framework Resource Properties

3.1. AggregatorServiceGroup Resource Properties

- `Entry`: This resource property publishes details of each registered resource, including both an EPR to the resource, the Aggregator Framework configuration information, and data from the sink.
- `RegistrationCount`: This resource property publishes registration load information (the total number of registrations since service startup and decaying averages)

4. Faults

The Aggregator Framework throws standard WS-ServiceGroup, WS-ResourceLifetime, and WS-ResourceProperties faults and does not define any new faults of its own.

5. WSDL and Schema Definition

- [AggregatorServiceGroup](#)³
- [AggregatorServiceGroupEntry](#)⁴
- [common types used by AggregatorServiceGroup and AggregatorServiceGroupEntry](#)⁵

Other relevant source files are the:

- [WSRF service group schema](#)⁶
- [WSRF resource lifetime schema](#)⁷
- MDS Usefulrp schema.

³ http://viewcvs.globus.org/viewcvs.cgi/ws-mds/aggregator/schema/mds/aggregator/aggregator_service_group_port_type.wsdl?revision=1.5&view=markup&pathrev=globus_4_2_branch

⁴ http://viewcvs.globus.org/viewcvs.cgi/ws-mds/aggregator/schema/mds/aggregator/aggregator_service_group_entry_port_type.wsdl?revision=1.6&view=markup&pathrev=globus_4_2_branch

⁵ http://viewcvs.globus.org/viewcvs.cgi/ws-mds/aggregator/schema/mds/aggregator/aggregator-types.xsd?revision=1.6&view=markup&pathrev=globus_4_2_branch

⁶ http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/servicegroup/sgw-2.wsdl?revision=1.2&view=markup&pathrev=globus_4_2_branch

⁷ http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/lifetime/rlw-2.wsdl?revision=1.2&view=markup&pathrev=globus_4_2_branch

Chapter 3. Configuring the WS MDS Index Service

Note

The aggregation source used to collect data can be changed from default, as detailed in the [Defining the Aggregator Sources](#) section below.

1. Configuration overview

For a basic installation, the Index Service itself does not need any configuration changes from default; a default Index Service is available and automatically "registers" with the following GT web services based resources to allow monitoring and discovery: [CAS], [RFT], and [GRAM4] (click the links for information about what data is sent and how to change it).

Note

Auto-registration is turned on by default in GT 4.2.1. See the per service links above for information about configuring this capability.

In order for information to appear in the Index Service, the source of that information must be registered to the Index Service. Information sources are registered using tools like [mds-servicegroup-add\(1\)](#). Each registration has a limited lifetime; **mds-servicegroup-add** should be left running in the background so that it can continue to refresh registrations. Depending on administration preference, it may be run on the same host as the index, on the same host as a member resource, or on any other host(s).

The Index Service is built on [Aggregator Framework](#) and can use any [Aggregator Sources Reference](#) to collect information. In the most common case, the index service uses the `QueryAggregatorSource` to gather resource property values from the registered resource using one of the three WS-Resource Properties operations to poll for information; the polling method used depends on the configuration element supplied in the registration content.

Two other aggregator sources are supplied with the distribution: the `SubscriptionAggregatorSource`, which gathers resource property values through subscription/notification, and the `ExecutionAggregatorSource`, which executes an external program to gather information.

2. Defining the Aggregator Sources

The aggregation sources used to collect data can be changed from default by editing the `aggregatorSources` parameter in the JNDI service configuration. See `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/jndi-config.xml`:

```
<resource name="configuration"
  type="org.globus.mds.aggregator.impl.AggregatorConfiguration">
  <resourceParams>
    <parameter>
      <name> factory</name>
      <value>org.globus_wsrf.jndi.BeanFactory</value>
    </parameter>
    <parameter>
```



```
<name>aggregatorSource</name>
<value>org.globus.mds.aggregator.impl.QueryAggregatorSource
      org.globus.mds.aggregator.impl.SubscriptionAggregatorSource
      org.globus.mds.aggregator.impl.ExecutionAggregatorSource
</value>
</parameter>
</resourceParams>
```

This parameter specifies one or more Java classes that may be used to collect data for the Index. By default it is set to use the `QueryAggregatorSource`, `SubscriptionAggregatorSource`, and `ExecutionAggregatorSource`. Details of these standard sources are in the [Aggregator Sources Reference](#).

Chapter 4. Graphical User Interface

There is no GUI specifically for the Index Service. The release contains WebMDS" which can be used to display monitoring information collected in an Index Service in a normal web browser.

Chapter 5. Configuring an executable to retrieve information

1. Interface introduction

The ExecutionAggregatorSource, which may be used by the Index Service, has a domain-specific interface (specifically, the inputs provided to and outputs expected from the executable program).

2. Syntax of the interface

The syntax of the execution source's domain-specific interface is described in [Chapter 6, Configuring Execution Aggregator Source](#).

Appendix A. Errors

Table A.1. WS MDS Index Service Error Messages

Error Code	Definition	Possible Solutions
error	what causes this	possible solutions
WS MDS is built on Java WS Core, please see Java WS Core Error Codes for more error code documentation.		

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.