

GT 4.0: Information Services: Aggregator Framework

GT 4.0: Information Services: Aggregator Framework

Table of Contents

Docs that relate to all MDS components: Key Concepts, Migrating Guide and Samples	vii
1. System Administrator's Guide	1
1. Introduction	1
2. Building and Installing	1
3. Configuring	1
4. Deploying	2
5. Testing	2
6. Security Considerations	2
7. Troubleshooting	3
8. Usage statistics collection by the Globus Alliance (4.0.5+)	3
2. User's Guide	4
1. Introduction	4
2. Command-line tools	4
3. Graphical user interfaces	5
4. Troubleshooting	5
5. Usage statistics collection by the Globus Alliance (4.0.5+)	5
3. Developer's Guide	6
1. Introduction	6
2. Before you begin	6
3. Architecture and design overview	7
4. Public interface	10
5. Usage scenarios	10
6. Tutorials	10
7. Debugging	11
8. Troubleshooting	11
9. Related Documentation	11
4. Fact Sheet	12
1. Brief component overview	12
2. Summary of features	12
3. Usability summary	12
4. Backward compatibility summary	12
5. Technology dependencies	13
6. Tested platforms	13
7. Associated standards	13
8. For More Information	14
5. Public Interface Guide	15
1. Semantics and syntax of APIs	15
2. Semantics and syntax of the WSDL	16
3. Command-line tools	17
4. Overview of Graphical User Interface	17
5. Semantics and syntax of domain-specific interface	18
6. Configuration interface	22
7. Environment variable interface	23
6. Quality Profile	24
1. Test coverage reports	24
2. Code analysis reports	24
3. Outstanding bugs	24
4. Bug Fixes	24
5. Performance reports	25
I. GT 4.0 WS MDS Aggregator Framework: Command Reference	26
mds-servicegroup-add	27

7. GT 4.0 WS MDS Aggregator Framework: Aggregator Sources Reference	32
1. Introduction	32
2. Types of Aggregator Sources in MDS4	32
3. Registering Aggregator Sources	33
4. Configuring Default Aggregator Sources	35
8. GT 4.0 WS MDS Aggregator Framework: Configuring the Execution Aggregator Source	36
1. Introduction	36
2. Registering	36
3. Configuration file: parameters for the execution aggregator source	38
4. Troubleshooting	39
5. Configuring the executable	39
9. GT 4.0 WS MDS Aggregator Framework: Configuring the Query Aggregator Source	41
1. Introduction	41
2. Registering	41
3. Configuration file: parameters for the query aggregator source	43
10. GT 4.0 WS MDS Aggregator Framework: Configuring the Subscription Aggregator Source	46
1. Introduction	46
2. Registering	46
3. Configuration file: parameters for the subscription aggregator source	48
11. 4.0.8 Release Notes	50
1. Introduction	50
2. Changes Summary	50
3. Bug Fixes	50
4. Known Problems	50
5. For More Information	50
12. 4.0.7 Release Notes	51
1. Introduction	51
2. Changes Summary	51
3. Bug Fixes	51
4. Known Problems	51
5. For More Information	51
13. 4.0.6 Release Notes	52
1. Introduction	52
2. Changes Summary	52
3. Bug Fixes	52
4. Known Problems	52
5. For More Information	52
14. 4.0.5 Release Notes	53
1. Introduction	53
2. Changes Summary	53
3. Bug Fixes	53
4. Known Problems	53
5. For More Information	53
15. 4.0.4 Release Notes	54
1. Introduction	54
2. Changes Summary	54
3. Bug Fixes	54
4. Known Problems	54
5. For More Information	54
16. 4.0.3 Release Notes	55
1. Introduction	55
2. Changes Summary	55
3. Bug Fixes	55
4. Known Problems	55

5. For More Information	55
17. 4.0.2 Release Notes	56
1. Introduction	56
2. Changes Summary	56
3. Bug Fixes	56
4. Known Problems	56
5. For More Information	56
18. 4.0.1 Release Notes	57
1. Introduction	57
2. Changes Summary	57
3. Bug Fixes	58
4. Known Problems	59
5. For More Information	59
19. 4.0.0 Release Notes	60
1. Component Overview	60
2. Feature Summary	60
3. Bug Fixes	60
4. Known Problems	61
5. Technology Dependencies	62
6. Tested Platforms	62
7. Backward Compatibility Summary	62
8. For More Information	62
GT 4.0 WS MDS Glossary	63

List of Tables

- 3.1. Standard aggregator sinks 9
- 3.2. Standard aggregator sources 10
- 5.1. Aggregator configuration parameters 20
- 4. Aggregator configuration parameters 30
- 7.1. Standard aggregator sources 33
- 7.2. Aggregator configuration parameters 35
- 8.1. Aggregator configuration parameters 38
- 9.1. Aggregator configuration parameters 43
- 10.1. Aggregator configuration parameters 48

Docs that relate to all MDS components: Key Concepts, Migrating Guide and Samples

- [Key Concepts](#)¹
- [Migrating Guide](#)²
- [Samples](#)³

¹ ../key-index.html

² ../WS_MDS_Migrating_Guide.html

³ ../WS_MDS_Samples.html

Chapter 1. GT 4.0 WS MDS Aggregator Framework: System Administrator's Guide

1. Introduction

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

Important

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

2. Building and Installing

The aggregator framework is built and installed as part of the standard [Globus Toolkit installation procedure](#)².

3. Configuring

The aggregator framework does not have its own service side configuration, although services which are based on the framework have their own service side configuration options (such as [MDS Index](#)³ and [MDS Trigger](#)⁴) which are documented in the per-service documentation.

Registrations to a working aggregator framework are configured for the [mds-servicegroup-add](#)⁵ tool. This tool takes an XML configuration file listing registrations, and causes those registrations to be made.

In general, configuration of aggregator services involves configuring the service to get information from one or more sources in a Grid. The mechanism for doing this is defined by (inherited from) the aggregator framework and described in this section.

3.1. Configuration overview

Each aggregator service has an associated ServiceGroup, which is used to keep track of configuration information and aggregated data. Configuring an Aggregating Service Group to perform a data aggregation is performed by adding a service group entry with the appropriate configuration information. This can be done from the command line using the `mds-servicegroup-add` command.

invocation. An AggregatorContent object is composed of two `xsd:any` arrays: AggregatorConfig, which contains registration parameters, and AggregatorData, which contains the actual collected data. Typically, the AggregatorConfig content is specified when the

¹ <http://www.globus.org/toolkit/docs/4.0/admin/docbook/>

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

³ <http://www.globus.org/toolkit/docs/4.0/info/index/>

⁴ <http://www.globus.org/toolkit/docs/4.0/info/trigger/>

⁵ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

- The `AggregatorConfig xsd:any` array is used to specify parameters that are to be passed to the underlying `AggregatorSource` when the `ServiceGroup add` method is invoked. These parameters are generally type-specific to the implementation of the `AggregatorSource` and/or `AggregatorSink` being used.
- The `AggregatorData xsd:any` array is used as the storage location for aggregated data that is the result of message deliveries to the `AggregatorSink`. Generally, the `AggregatorData` parameter of the `AggregatorContent` is not populated when the `ServiceGroup add` method is invoked, but rather is populated by message delivery from the `AggregatorSource`.

3.2. Syntax of the interface

3.2.1. Configuring the Aggregator Sources

The following links provide information for configuring the three types of aggregator sources provided by the Globus Toolkit:

- [Configuring the Execution Aggregator Source](#)⁶
- [Configuring the Query Aggregator Source](#)⁷
- [Configuring the Subscription Aggregator Source](#)⁸

3.2.2. Configuring the Aggregator Sink

An aggregator sink may require sink-specific configuration (the MDS *Trigger service* requires sink-specific configuration; the MDS *Index service* does not). See the documentation for the specific *aggregator service* being used for details on sink-specific documentation.

4. Deploying

This component is deployed as part of the standard toolkit installation.

5. Testing

The aggregator framework is a software framework used to create services. To test that the aggregator framework is working, deploy and test a service (such as the [Index Service](#)⁹).

6. 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.

⁶ http://www.globus.org/toolkit/docs/4.0/info/aggregator/Execution_Aggregator_Source.html

⁷ http://www.globus.org/toolkit/docs/4.0/info/aggregator/Query_Aggregator_Source.html

⁸ http://www.globus.org/toolkit/docs/4.0/info/aggregator/Subscription_Aggregator_Source.html

⁹ <http://www.globus.org/toolkit/docs/4.0/info/index/>

7. Troubleshooting

*Problem: I was able to successfully register an aggregator entry with **mds-servicegroup-add**, but the aggregator isn't collecting data for the registration.*

Solution: The fact that the registration was successful does not mean that there are no errors in the registration parameters. Verify that details such as resource EPRs, resource property names, and queries are accurate, and check the container logs for the *aggregator service* and (if applicable) the remote service for more information.

*Problem: I was able to successfully register an aggregator entry with **mds-servicegroup-add**, and the aggregator collected information for this entry for a while, but then the entry disappeared.*

Solution: make sure that **mds-servicegroup-add** is still running. Registrations time out; **mds-servicegroup-add** refreshes them periodically.

8. Usage statistics collection by the Globus Alliance (4.0.5+)

Starting with 4.0.5, the following usage statistics are sent by default in a UDP packet each time a registration is made to an Index Service or Trigger Service:

- Service name
- Total number of registrations
- Number of current registrations
- Time that the aggregator resource was created

If you wish to disable this feature, please see the Java WS Core System Administrator's Guide section on [Usage Statistics Configuration](#)¹⁰ for instructions.

Also, please see our [policy statement](#)¹¹ on the collection of usage statistics.

¹⁰ http://www.globus.org/toolkit/docs/4.0/common/javawscore/admin-index.html#s-javawscore-Interface_Config_Frag-usageStatisticsTargets

¹¹ http://www.globus.org/toolkit/docs/4.0/Usage_Stats.html

Chapter 2. GT 4.0 WS MDS Aggregator Framework: User's Guide

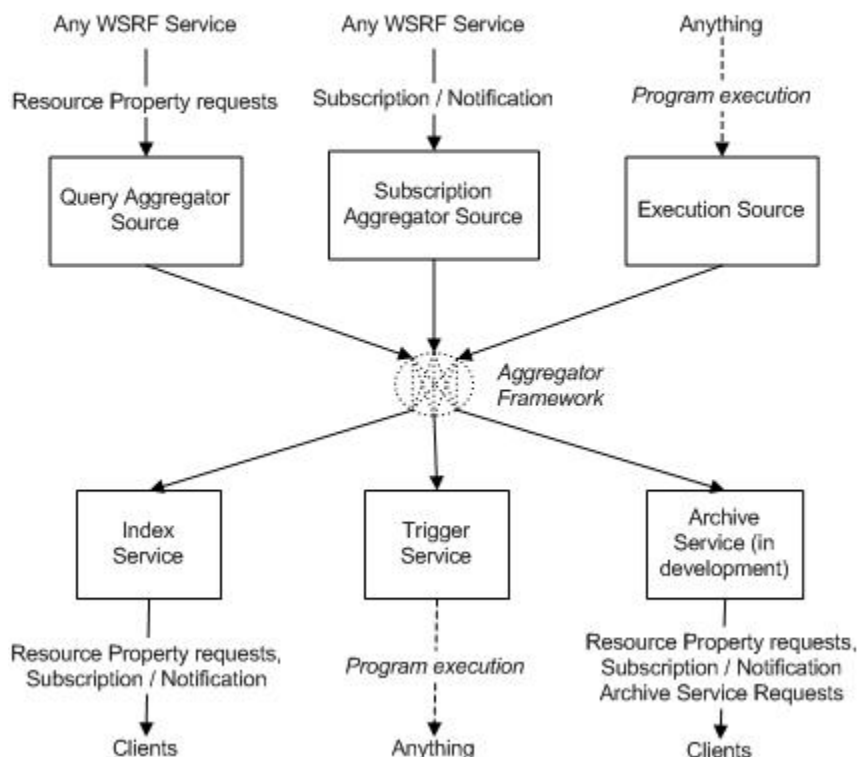
1. Introduction

This component does not generally have end-users.

The WS MDS Aggregator Framework is the software framework on which WS MDS services (currently, the WS MDS *Index* and WS MDS *Trigger* services) are built. The aggregator framework collects data from an *aggregator source* and sends that data to an *aggregator sink* for processing. Aggregator sources distributed with the Globus Toolkit include modules that query service data, acquire data through subscription/notification, and execute programs to generate data. Aggregator sinks include modules that implement the WS MDS Index service interface and the WS MDS Trigger service interface.

In general, end-users will *not* interact directly with the aggregator framework; instead, they will interact with clients specific to the aggregator sources and aggregator sinks.

Information Flow in WS-MDS



2. Command-line tools

See [Aggregator Command Reference](#) for details.

3. Graphical user interfaces

There is no GUI specifically for the aggregator framework. The release contains [WebMDS](#)¹ which can be used to display monitoring information in a web browser. Specifically, it can be directed at services based on the aggregator framework to display information about resources registered to the aggregator framework.

4. Troubleshooting

End-users do not generally interact directly with the aggregator framework.

General troubleshooting information can be found in the [GT 4.0 Java WS Core User's Guide](#)².

5. Usage statistics collection by the Globus Alliance (4.0.5+)

Starting with 4.0.5, the following usage statistics are sent by default in a UDP packet each time a registration is made to an Index Service or Trigger Service:

- Service name
- Total number of registrations
- Number of current registrations
- Time that the aggregator resource was created

If you wish to disable this feature, please see the Java WS Core System Administrator's Guide section on [Usage Statistics Configuration](#)³ for instructions.

Also, please see our [policy statement](#)⁴ on the collection of usage statistics.

¹ <http://www.globus.org/toolkit/docs/4.0/info/webmds/index.html>

² <http://www.globus.org/toolkit/docs/4.0/common/javawscore/user-index.html#troubleshooting>

³ http://www.globus.org/toolkit/docs/4.0/common/javawscore/admin-index.html#s-javawscore-Interface_Config_Frag-usageStatisticsTargets

⁴ http://www.globus.org/toolkit/docs/4.0/Usage_Stats.html

Chapter 3. GT 4.0 WS MDS Aggregator Framework: Developer's Guide

1. Introduction

The aggregator framework allows pluggable data sources and sinks to be written and connected together. Generally a source collects data from or about a particular grid resource, and passes it to a sink which does something interesting with it.

The aggregator sinks supplied with the toolkit implement the [WS MDS Index Service](#)¹ and [WS MDS Trigger Service](#)². The *aggregator sources* supplied with the toolkit collect information using resource property queries, subscription/notification, and execution of external programs.

This document describes the programmatic interfaces to the aggregator framework. See also general Globus Toolkit [coding guidelines](#)³ and [GT 4.0 best practices](#)⁴.

2. Before you begin

2.1. Feature summary

Features new in release GT 4.0

- Ported to use WSRF mechanisms (previously used OGSF).
- Additional sources which collect information by polling and by execution of local scripts.
- Management of aggregations is now performed over the wire through WS ServiceGroup APIs.

Other Supported Features

- Collects information from grid resources using pluggable aggregation sources.
- Delivers collected information to pluggable sinks.
- Manages creation and destruction of individual aggregation registrations.

2.2. Tested platforms

Tested Platforms for WS MDS Aggregator Framework

- Linux on i386
- Windows XP

¹ <http://www.globus.org/toolkit/docs/4.0/info/index>

² <http://www.globus.org/toolkit/docs/4.0/info/trigger>

³ http://www.globus.org/toolkit/docs/development/coding_guidelines.html

⁴ http://www.globus.org/toolkit/docs/4.0/best_practices.html

2.3. Backward compatibility summary

Protocol changes since GT version 3.2

- The aggregator framework is a complete reimplementaion of the MDS3 aggregator framework using WSRF rather than OGSi protocols.
- No wireside compatibility with MDS3 aggregator framework.
- Architectural similarity should make porting straightforward.

API changes since GT version 3.2

- APIs entirely rewritten, so no API compatibility.
- Architectural similarity should make porting straightforward.

Exception changes since GT version 3.2

- See API changes above.

Schema changes since GT version 3.2

- Registration interface uses WSRF rather than OGSi schemas.
- New per-source and per-sink configuration schemas.

2.4. Technology dependencies

Aggregator Framework depends on the following GT components:

- Java WS Core

Aggregator Framework depends on the following 3rd party software:

- None

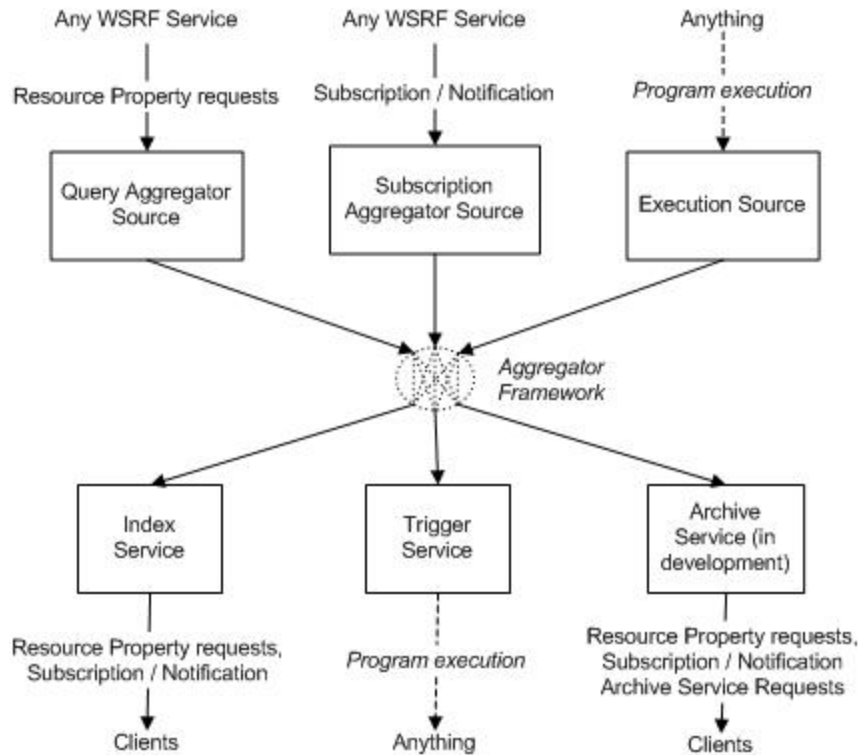
2.5. Security considerations

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.

3. Architecture and design overview

The WS MDS Aggregator Framework is the software framework on which WS MDS services are built. The aggregator framework collects data from an *aggregator source* and sends that data to an *aggregator sink* for processing. Aggregator sources distributed with the Globus Toolkit include modules that query resource properties, acquire data through subscription/notification, and execute programs to generate data. Another way of describing the Aggregator Framework is that it is designed to facilitate the collecting of information from or about WS-Resources via plugin aggregator sources and the feeding of that information to plugin aggregator sinks, which can then perform actions such as re-publishing, logging, or archiving the information.

Information Flow in WS-MDS



Aggregators work on a type of service group called an **AggregatorServiceGroupRP**. Resources may be *registered* to an **AggregatorServiceGroupRP** 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 appropriate aggregation source and sinks will be informed; the aggregator source will begin collecting data and inserting it into the corresponding service group entry, 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.

3.1. Standard aggregator sinks

The aggregator sinks distributed with the toolkit (`org.globus.mds.aggregator.impl.ServiceGroupEntryAggregatorSink` and `org.globus.mds.trigger.impl.TriggerResource`) are described in the following table.

Table 3.1. Standard aggregator sinks

Aggregator Sink	Service Implemented	Description
ServiceGroupEntry- AggregatorSink	Index Service	The servicegroup sink (used by the <i>Index Service</i>) publishes received data as content in the <code>AggregatingServiceGroup</code> entry used to manage the registration. This data can therefore be retrieved by querying the index for its 'entries' resource property.
TriggerResource	Trigger Service	The <i>Trigger Service</i> provides an aggregator sink which receives data, applies tests to that data, and if the tests match, runs a specified executable. See the trigger service ⁵ documentation for more information.

3.2. Standard aggregator sources

The aggregator sources supplied with the toolkit collect information using resource property queries (query sources), subscription/notification (subscription sources), and execution of external programs (execution sources).

The aggregator sources supplied with the Globus Toolkit are listed in the following table.



Note

All aggregator sources listed in this table are in the `org.globus.mds.aggregator.impl` package, so for example the aggregator source listed as `QueryAggregatorSource` is actually `org.globus.mds.aggregator.impl.QueryAggregatorSource`

⁵ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/trigger/>

Table 3.2. Standard aggregator sources

Aggregator Source	Description
QueryAggregatorSource	<p>The query source collects information from a registered resource by using WS-Resource Properties polling mechanisms:</p> <ul style="list-style-type: none"> • <code>GetResourcePropertyPollType</code>; requests a single Resource Property from the remote resource. • <code>GetMultipleResourcePropertiesPollType</code>; requests multiple Resource Properties from the remote resource. • <code>QueryResourcePropertiesPollType</code>; requests a query be executed against the Resource Property Set of the remote resource. <p>Polls are made periodically, with both the period and target Resource Properties specified in the registration message.</p>
SubscriptionAggregator-Source	<p>The subscription source collects information from a registered resource using WS-Notification mechanisms. Data is delivered when property values change, rather than periodically.</p>
ExecutionAggregator-Source	<p>The execution source collects information about (not necessarily from) a registered resource by execution of a local executable, which is passed as input the identity of the registered resource. Details of the interface between the execution source and local executables are in Configuring the Execution Aggregator Source⁶.</p>

4. Public interface

The semantics and syntax of the APIs and WSDL for the component, along with descriptions of domain-specific structured interface data, can be found in the [public interface guide](#)⁷.

5. Usage scenarios

5.1. Creating WS MDS services

The aggregator framework is used to create MDS services by linking an *aggregator source* (a java class that implements the AggregatorSource interface to collect data) to an *aggregator sink* (a java class that implements the AggregatorSink interface to process data, e.g., by providing a service interface for it). The AggregatorSource and AggregatorSink interfaces are documented in the aggregator [Public Interface Guide](#)⁸.

6. Tutorials

Use of the index service (based on the WS MDS Aggregator Framework) is covered in the [Build a Grid Service Tutorial \(GlobusWORLD 2005\)](#)⁹.

⁶ http://www.globus.org/toolkit/docs/4.0/info/aggregator/Execution_Aggregator_Source.html

⁷ [WS_MDS_Aggregator_Public_Interfaces.html](#)

⁸ [WS_MDS_Aggregator_Public_Interfaces.html](#)

⁹ <http://www.globus.org/toolkit/tutorials/BAS/>

7. Debugging

See the [Debug section of the Java WS Core Developer's Guide](#)¹⁰ for general information on logging, including which files to edit to set logging properties.

To turn on debug logging for the Aggregator framework, add the line:

```
log4j.category.org.globus.mds.aggregator=DEBUG
```

to the appropriate properties file.

8. Troubleshooting

General troubleshooting information can be found in the [GT 4.0 Java WS Core Developer's Guide](#)¹¹.

9. Related Documentation

Specifications for resource properties, service groups, and subscription/notification are available at <http://www.globus.org/wsr/>.

¹⁰ [../common/javawscore/developer-index.html#debugging](#)

¹¹ [../common/javawscore/developer-index.html#troubleshooting](#)

Chapter 4. GT 4.0 WS MDS Aggregator Framework: Fact Sheet

1. Brief component overview

The Aggregator Framework is the software framework on which WS MDS services (currently, the WS MDS *Index* and WS MDS *Trigger* services) are built. The aggregator framework collects data from an *aggregator source* and sends that data to an *aggregator sink* for processing. Aggregator sources distributed with the Globus Toolkit include modules that query service data, acquire data through subscription/notification, and execute programs to generate data. Aggregator sinks include modules that implement the WS MDS Index service interface and the WS MDS Trigger service interface.

2. Summary of features

Features new in release GT 4.0

- Ported to use WSRF mechanisms (previously used OGSF).
- Additional sources which collect information by polling and by execution of local scripts.
- Management of aggregations is now performed over the wire through WS ServiceGroup APIs.

Other Supported Features

- Collects information from grid resources using pluggable aggregation sources.
- Delivers collected information to pluggable sinks.
- Manages creation and destruction of individual aggregation registrations.

3. Usability summary

Usability improvements for WS MDS Aggregator Framework:

- While the aggregator concept is not new to GT4, the MDS4 aggregator implementation is. In comparison to GT3, the aggregator framework has been rewritten to support the WSRF model and generalized to support a more flexible plug-in environment.
- Also different from the GT3 aggregator, service to service aggregation sessions are now established exclusively through Service Group registrations, and the lifetime of these aggregations is managed via WSRF Resource Lifetime functions.

4. Backward compatibility summary

Protocol changes since GT version 3.2

- The aggregator framework is a complete reimplemention of the MDS3 aggregator framework using WSRF rather than OGSF protocols.

- No wireside compatibility with MDS3 aggregator framework.
- Architectural similarity should make porting straightforward.

API changes since GT version 3.2

- APIs entirely rewritten, so no API compatibility.
- Architectural similarity should make porting straightforward.

Exception changes since GT version 3.2

- See API changes above.

Schema changes since GT version 3.2

- Registration interface uses WSRF rather than OGSF schemas.
- New per-source and per-sink configuration schemas.

5. Technology dependencies

Aggregator Framework depends on the following GT components:

- Java WS Core

Aggregator Framework depends on the following 3rd party software:

- None

6. Tested platforms

Tested Platforms for WS MDS Aggregator Framework

- Linux on i386
- Windows XP

7. Associated standards

Associated standards for WS MDS Aggregator Framework:

- WS-ResourceProperties (WSRF-RP)
- WS-ResourceLifetime (WSRF-RL)
- WS-ServiceGroup (WSRF-SG)
- WS-BaseNotification
- WS-Topics

8. For More Information

Click [here](#)¹ for more information about this component.

¹ index.html

Chapter 5. GT 4.0 WS MDS Aggregator Framework: Public Interface Guide

1. Semantics and syntax of APIs

1.1. Programming Model Overview

The aggregator framework module consists of an Aggregating ServiceGroup framework which supports plugins as detailed below, as well as a number of standard plugins.

1.2. The Aggregating ServiceGroup framework

The aggregating servicegroup framework is designed to facilitate the collecting of information from or about WS-Resources (via plugin *aggregator sources*) and the feeding of that information to plugin aggregator sinks.

The framework provides for over-the-wire management of the list of registered resources (through a WS-ServiceGroup interface) and a Java API for connecting sources and sinks together.

In general (although this is not a hard requirement), aggregator sinks will be tied into a specific service implementation, while aggregator sources are more independent. (For example, the trigger and index services act as sinks)

1.3. The standard plugins

A number of standard aggregator sources are provided, which implement the aggregator source API. These provide for collecting information from/about a WS-Resource by:

- WS-ResourceProperties poll operations
- WS-Notification subscription
- Execution of arbitrary executables

See [Aggregator Sources Reference](#)¹ for more information about standard aggregator sources for GT 4.0.

1.4. Component API

There are two main Java interfaces in the aggregator framework.

- [AggregatorSink](#)² - which is implemented by sinks that can receive data from the aggregator framework.
- [AggregatorSource](#)³ - which is implemented by sources that can feed data into the aggregator framework.

In addition, the AggregatorContent class is used when configuring an aggregator service programmatically, and to represent the data published in the aggregator's `Entry` resource property. All aggregator classes and interfaces are documented in the [aggregator Java API documentation](#)⁴

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Sources_Reference.html

² http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_aggregator/org/globus/mds/aggregator/impl/AggregatorSink.html

³ http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_aggregator/org/globus/mds/aggregator/impl/AggregatorSource.html

⁴ http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_aggregator/

2. Semantics and syntax of the WSDL

2.1. Protocol overview

The aggregator framework builds on the [WS-ServiceGroup](#)⁵ and [WS-ResourceLifetime](#)⁶ 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](#)⁷ for source information and per-service documentation for sink information - for example the [Index Service](#)⁸ and the [Trigger Service](#)⁹.)

2.2. Operations

2.2.1. 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 [Aggregator Administrator's Guide](#)¹⁰).

2.2.2. AggregatorServiceGroupEntry

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

2.3. Resource properties

2.3.1. 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.

⁵ http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/servicegroup/WS-ServiceGroup.wsdl?rev=1.9&only_with_tag=globus_4_0_0

⁶ http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/lifetime/WS-ResourceLifetime.wsdl?rev=1.11&only_with_tag=globus_4_0_0

⁷ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Sources_Reference.html

⁸ <http://www.globus.org/toolkit/docs/4.0/info/index/>

⁹ <http://www.globus.org/toolkit/docs/4.0/info/trigger/>

¹⁰ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/admin-index.html>

- `RegistrationCount`: This resource property publishes registration load information (the total number of registrations since service startup and decaying averages)

2.4. Faults

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

2.5. WSDL and Schema Definition

- [AggregatorServiceGroup](#)¹¹
- [AggregatorServiceGroupEntry](#)¹²
- [common types used by AggregatorServiceGroup and AggregatorServiceGroupEntry](#)¹³

Other relevant source files are the:

- [WSRF service group schema](#)¹⁴
- [WSRF resource lifetime schema](#)¹⁵
- MDS Usefulrp schema.

3. Command-line tools

Please see the [Aggregator Command Reference](#).

4. Overview of Graphical User Interface

There is no GUI specifically for the aggregator framework. The release contains [WebMDS](#)¹⁶ which can be used to display monitoring information in a web browser. Specifically, it can be directed at services based on the aggregator framework to display information about resources registered to the aggregator framework.

¹¹ http://viewcvs.globus.org/viewcvs.cgi/ws-mds/aggregator/schema/mds/aggregator/aggregator_service_group_port_type.wsdl?rev=1.1&only_with_tag=globus_4_0_0&content-type=text/vnd.viewcvs-markup

¹² http://viewcvs.globus.org/viewcvs.cgi/ws-mds/aggregator/schema/mds/aggregator/aggregator_service_group_entry_port_type.wsdl?rev=1.2&only_with_tag=globus_4_0_0&content-type=text/vnd.viewcvs-markup

¹³ http://viewcvs.globus.org/viewcvs.cgi/ws-mds/aggregator/schema/mds/aggregator/aggregator-types.xsd?rev=1.1&only_with_tag=globus_4_0_0&content-type=text/vnd.viewcvs-markup

¹⁴ http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/servicegroup/WS-ServiceGroup.wsdl?rev=1.9&only_with_tag=globus_4_0_0

¹⁵ http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/lifetime/WS-ResourceLifetime.wsdl?rev=1.11&only_with_tag=globus_4_0_0

¹⁶ <http://www.globus.org/toolkit/docs/4.0/info/webmds/index.html>

5. Semantics and syntax of domain-specific interface

5.1. Writing executable to be called by execution aggregator source

5.1.1. Introduction

The execution aggregation source provides a way to aggregate data (arbitrary XML information) about a registered resource using an arbitrary local executable (such as an external script). The executable will be passed registration information as parameters and is expected to output the gathered data, as detailed below.

A basic example of the use of this API is described in the [ping test example for the aggregator execution source](#)¹⁷

The execution aggregation source will periodically execute an identified executable. The identity of the executable and the frequency with which it is to run are specified in the registration message.

5.1.2. Registering

To register resources:

- Create a configuration file in XML that specifies registrations. See `$GLOBUS_LOCATION/etc/globus_ws-rf_mds_aggregator/example-aggregator-registration.xml` for several specific examples.
- Run `mds-servicegroup-add`¹⁸ to perform the registrations specified in that configuration file.

The file consists of an optional `defaultServiceGroupEPR`, an optional `defaultRegistrantEPR`, and then one or more `ServiceGroupRegistrationParameters` blocks, each of which represents one registration.

The general syntax of the configuration file is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceGroupRegistrations
  xmlns="http://mds.globus.org/servicegroup/client">

  // An optional default service group EPR.
  <defaultServiceGroupEPR>
    // Default service group EPR
  </defaultServiceGroupEPR>

  // An optional default registrant EPR.
  <defaultRegistrantEPR>
    // Default registrant EPR
  </defaultRegistrantEPR>

  // An optional default security descriptor file.
```

¹⁷ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/developer/pingtest.html>

¹⁸ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

```
<defaultSecurityDescriptorFile>
  // Path name of default security descriptor file
</defaultSecurityDescriptorFile>

// One or more service group registration blocks:

<ServiceGroupRegistrationParameters>
  <ServiceGroupEPR>
    // EPR of the service group to register to
  </ServiceGroupEPR>
  <RegistrantEPR>
    // EPR of the entity to be monitored.
  </RegistrantEPR>
  <InitialTerminationTime>
    // Initial termination time
  </InitialTerminationTime>
  <RefreshIntervalSecs>
    // Refresh interval, in seconds
  </RefreshIntervalSecs>
  <Content type="agg:AggregatorContent">
    // Aggregator-source-specific configuration parameters
  </Content>
</ServiceGroupRegistrationParameters>

</ServiceGroupRegistrations>
```

The following table describes the different blocks of the file and any parameters:

Table 5.1. Aggregator configuration parameters

defaultServiceGroupEPR block	The provides a convenient way to register a number of resources to a single service group -- for example, if you wish to register several resources to your default VO index, you can specify that index as the default service group and omit the ServiceGroupEPR blocks from each ServiceGroupRegistrationParameters block.
defaultRegistrantEPR	The provides a convenient way to register a single resource to several service groups -- for example, if you wish to register your local GRAM server to several index servers, you can specify your GRAM server as the default registrant and omit the RegistrantEPR blocks from each ServiceGroupRegistrationParameters block.
defaultSecurityDescriptorFile	Simply the path to the <u>security descriptor file</u> ¹⁹ .
ServiceGroupRegistrationParameters	Each ServiceGroupRegistrationParameters block specifies the parameters used to register a resource to a service group. The parameters specified in this block are:
ServiceGroupEPR	The EPR of the service group to register to. This parameter may be omitted if a defaultServiceGroupEPR block is specified; in this case, the value of defaultServiceGroupEPR will be used instead.
RegistrantEPR	The EPR of the resource to register. This parameter may be omitted if a defaultRegistrantEPR block is specified; in this case, the value of defaultRegistrantEPR will be used instead.
InitialTerminationTime	The initial termination time of this registration (this may be omitted). If the initial termination time is omitted, then the <u>mds-servicegroup-add</u> ²⁰ sets the initial termination time to the current wall time plus 2 times that of the specified RefreshIntervalSecs parameter.
RefreshIntervalSecs	The refresh interval of the registration, in seconds. The <u>mds-servicegroup-add</u> ²¹ will attempt to refresh the registration according to this interval, by default incrementing the termination time of the registration by 2 times this interval for every successful refresh. If at any point the termination time for the registration expires the registration will be subject to removal within a maximum of 5 minutes.
Content	Aggregator-source-specific registration parameters. The content blocks for the various aggregator sources are described in detail in the following sections.

5.1.3. Configuration file: parameters for the execution aggregator source

The configuration block for ExecutionAggregatorSource (inside the Content block) looks like this:

```
<Content xsi:type="agg:AggregatorContent"
  xmlns:agg="http://mds.globus.org/aggregator/types">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
    <agg:ExecutionPollType>
      <agg:PollIntervalMillis>interval_in_ms</agg:PollIntervalMillis>
      <agg:ProbeName>dummy_namespace:probe_name</agg:ProbeName>
    </agg:ExecutionPollType>
  </agg:AggregatorConfig>
</Content>
```

¹⁹ http://www.globus.org/toolkit/docs/4.0/security/authzframe/security_descriptor.html

²⁰ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

²¹ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

```

    </agg:ExecutionPollType>
  </agg:AggregatorConfig>
  <agg:AggregatorData/>
</Content>

```

where:

PollIntervalMillis This parameter is the poll refresh period in milliseconds.

ProbeName This parameter specifies name of the probe to run. This probe is defined in the `jndi-config.xml` file for the service being configured (for example, the file for the MDS Index service is `$GLOBUS_LOCATION/etc/globus_wsrp_mds_index_jndi-config.xml`). An `executableMappings` parameter should be defined within this file to map probe names to executable names. For example, this maps the probe names `aggr-test` and `pingexec` to the executables called `aggregator-exec-test.sh` and `example-ping-exec`, respectively. All executables are presumed to be in the directory `$GLOBUS_LOCATION/libexec/aggrexec`.

```

<resource name="configuration"
          type="org.globus.mds.aggregator.impl.AggregatorConfigurati
  <resourceParams>
    // ...
    <parameter>
      <name>executableMappings</name>
      <value>aggr-test=aggregator-exec-test.sh, pingexec=example-ping-
    </parameter>
  </resourceParams>
</resource>

```

5.1.4. Troubleshooting

If you've properly configured and registered your script for execution but are getting errors from the container because it cannot find the specified script, there are two likely causes.

First, make sure that your script/program is executable and is located in the `$GLOBUS_LOCATION/libexec/aggrexec` directory. When it's specified in the configuration mentioned above, only specify the name of the script/program, without any qualification or path. For example, using the `ProbeName` as `test-script` will be specifying the file `$GLOBUS_LOCATION/libexec/aggrexec/test-script` script.

Next, make sure that you have correctly created an `executableMappings` definition in the appropriate `jndi-config.xml` file.

5.1.5. Configuring the executable

5.1.5.1. Name of executable

The executable to run will be `$GLOBUS_LOCATION/libexec/aggrexec/<scriptname>` with `scriptname` supplied by the `ProbeName` parameter in the configuration file.

5.1.5.2. Input to executable

Information about the registration will be supplied as command line parameters and on `stdin`.

A single command line parameter will be supplied to the executable. This will be the URL from the EPR of the registered service.

Two XML documents will be sent to stdin, in sequence:

1. The first document will be the full EPR to the registered service.
2. The second document will be the `AggregatorConfig` block from the registration message (configuration file).

5.1.5.3. Output from executable

The executable must output a well-formed XML document to stdout. This output document will be delivered into the aggregator framework.

6. Configuration interface

The aggregator framework does not have its own service side configuration, although services which are based on the framework have their own service side configuration options (such as [MDS Index](#)²² and [MDS Trigger](#)²³) which are documented in the per-service documentation.

Registrations to a working aggregator framework are configured for the `mds-servicegroup-add`²⁴ tool. This tool takes an XML configuration file listing registrations, and causes those registrations to be made.

In general, configuration of aggregator services involves configuring the service to get information from one or more sources in a Grid. The mechanism for doing this is defined by (inherited from) the aggregator framework and described in this section.

6.1. Configuration overview

Each aggregator service has an associated `ServiceGroup`, which is used to keep track of configuration information and aggregated data. Configuring an Aggregating Service Group to perform a data aggregation is performed by adding a service group entry with the appropriate configuration information. This can be done from the command line using the `mds-servicegroup-add` command.

invocation. An `AggregatorContent` object is composed of two `xsd:any` arrays: `AggregatorConfig`, which contains registration parameters, and `AggregatorData`, which contains the actual collected data. Typically, the `AggregatorConfig` content is specified when the

- The `AggregatorConfig` `xsd:any` array is used to specify parameters that are to be passed to the underlying `AggregatorSource` when the `ServiceGroup` `add` method is invoked. These parameters are generally type-specific to the implementation of the `AggregatorSource` and/or `AggregatorSink` being used.
- The `AggregatorData` `xsd:any` array is used as the storage location for aggregated data that is the result of message deliveries to the `AggregatorSink`. Generally, the `AggregatorData` parameter of the `AggregatorContent` is not populated when the `ServiceGroup` `add` method is invoked, but rather is populated by message delivery from the `AggregatorSource`.

²² <http://www.globus.org/toolkit/docs/4.0/info/index/>

²³ <http://www.globus.org/toolkit/docs/4.0/info/trigger/>

²⁴ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

6.2. Syntax of the interface

6.2.1. Configuring the Aggregator Sources

The following links provide information for configuring the three types of aggregator sources provided by the Globus Toolkit:

- [Configuring the Execution Aggregator Source](#)²⁵
- [Configuring the Query Aggregator Source](#)²⁶
- [Configuring the Subscription Aggregator Source](#)²⁷

6.2.2. Configuring the Aggregator Sink

An aggregator sink may require sink-specific configuration (the MDS *Trigger service* requires sink-specific configuration; the MDS *Index service* does not). See the documentation for the specific *aggregator service* being used for details on sink-specific documentation.

7. Environment variable interface

There are no environment variables specific to the aggregator framework.

²⁵ http://www.globus.org/toolkit/docs/4.0/info/aggregator/Execution_Aggregator_Source.html

²⁶ http://www.globus.org/toolkit/docs/4.0/info/aggregator/Query_Aggregator_Source.html

²⁷ http://www.globus.org/toolkit/docs/4.0/info/aggregator/Subscription_Aggregator_Source.html

Chapter 6. GT 4.0 WS MDS Aggregator Framework: Quality Profile

1. Test coverage reports

- None available at this time.

2. Code analysis reports

- None available at this time.

3. Outstanding bugs

- [2807: execution aggregator source junit test needs some work](#)¹
- [3150: AggregatingServiceGroup* needs to provide Resource.Remove...](#)²
- [3177: AggregatingServiceGroup EntrySweeper interval needs to be...](#)³
- [All open aggregator bug reports and enhancement requests](#)⁴

4. Bug Fixes

- [Bug 2432: Aggregator sample config file should have tls-style urls](#)⁵
- [Bug 2388: rejected add\(\) leaves incomplete state behind which then upsets the sweeper](#)⁶
- [Bug 2384: remove stack trace from execution source error](#)⁷
- [Bug 2377: support multiple aggregator sources in one aggregator](#)⁸
- [Bug 2157: Aggregator entries do not implement WS-Resource Lifetime resource properties](#)⁹
- [Bug 2145: query source waits entire reg period before making first request](#)¹⁰
- [Bug 2105: flatten schema in CVS](#)¹¹

¹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2807

² http://bugzilla.globus.org/globus/show_bug.cgi?id=3150

³ http://bugzilla.globus.org/globus/show_bug.cgi?id=3177

⁴ http://bugzilla.globus.org/globus/buglist.cgi?short_desc_type=allwordssubstr&short_desc=&product=MDS&component=wsrf_aggregator&long_desc_type=allwordssubstr&long_desc=&bug_file_loc_type=allwordssubstr&bug_file_loc=&bug_status=NEW&bug_status=AS-SIGNED&bug_status=REOPENED&emailtype1=substring&email1=&emailtype2=substring&email2=&bugidtype=include&bug_id=&votes=&changedin=&chfieldfrom=&chfieldto=Now&chfieldvalue=&cmdtype=doit&newqueryname=&order=Reuse+same+sort+as+last+time&field0-0=noop&type0-0=noop&value0-0-0=

⁵ http://bugzilla.globus.org/globus/show_bug.cgi?id=2432

⁶ http://bugzilla.globus.org/globus/show_bug.cgi?id=2388

⁷ http://bugzilla.globus.org/globus/show_bug.cgi?id=2384

⁸ http://bugzilla.globus.org/globus/show_bug.cgi?id=2377

⁹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2157

¹⁰ http://bugzilla.globus.org/globus/show_bug.cgi?id=2145

¹¹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2105

- [Bug 2097: IndexDefaultWidgetTest registers same widget many times](#)¹²
- [Bug 2087: getresourceproperty poll source does not tolerate whitespace in RP name](#)¹³
- [2082: index and aggregator stub generation is generating metrics stubs](#)¹⁴
- [2191: AggregatingServiceGroupEntries do not implement Service Group EPR properly](#)¹⁵
- [All fixed aggregator framework bugs and enhancement requests](#)¹⁶
- [2339: Registered services connection refused messages](#)¹⁷
- [2398: index service does not appear to have wsrf-query support \(any more\)](#)¹⁸
- [2850: MDS4 logs to INFO when DEBUG should be used](#)¹⁹
- [2958: WS-MDS Aggregator Junit test execution has no results](#)²⁰
- [2963: need to create libexec/aggrexec dir as part of MDS install](#)²¹

5. Performance reports

- None available at this time.

¹² http://bugzilla.globus.org/globus/show_bug.cgi?id=2097

¹³ http://bugzilla.globus.org/globus/show_bug.cgi?id=2087

¹⁴ http://bugzilla.globus.org/globus/show_bug.cgi?id=2082

¹⁵ http://bugzilla.globus.org/globus/show_bug.cgi?id=2191

¹⁶ <http://bugzilla.globus.org/globus/buglist.cgi?regetlastlist=1>

¹⁷ http://bugzilla.globus.org/globus/show_bug.cgi?id=2339

¹⁸ http://bugzilla.globus.org/globus/show_bug.cgi?id=2398

¹⁹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2850

²⁰ http://bugzilla.globus.org/globus/show_bug.cgi?id=2958

²¹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2963

GT 4.0 WS MDS Aggregator Framework: Command Reference

Name

`mds-servicegroup-add` -- Registering grid resources to MDS services such as the Index and Trigger services

`mds-servicegroup-add`

Tool description

mds-servicegroup-add creates a set of registrations to a WS-ServiceGroup and periodically renews those registrations. It is intended primarily for registering grid resources to MDS services such as the Index and Trigger services.

The tool can be deployed at the aggregating service, at resource services, or at any other location.

This allows registrations to be configured by the administrator of the aggregating service, or by the administrator of resources, by a third party, or by some combination of those.

Registrations are defined in an XML configuration file, which is documented here: [Registering with mds-servicegroup-add](#).

For an example using an Index Service, see [Simple usage for the Index Service](#).

And remember to note the section on [Limitations](#).

Command syntax

The basic syntax for **mds-servicegroup-add** is:

```
mds-servicegroup-add -s http://foo [options] config.xml
```

where:

<code>-s http://foo</code>	This dummy option is required but ignored. A URL to the service group against which the <code>mds-servicegroup-add</code> request will be executed. This command line argument is a required argument, however it is only necessary that this URL argument evaluate to a valid EPR in the case that there are no suitable target service group EPRs present in the configuration file . Any end point references found in the configuration file will automatically override the EPR specified on the command-line.
<code>-a</code>	By default, mds-servicegroup-add will attempt to make an authenticated connection to each service group. This option is used to specify anonymous connections (and to prevent mds-servicegroup-add from failing if you don't have a valid Grid credential).
<code>-z auth_type</code>	Specify an authorization type: <code>self</code> Fail if the server's identity is different from the user's identity. <code>host</code> Fail if the server does not have a valid server certificate. <code>none</code> Continue regardless of the server's identity.
<code>config.xml</code>	Path to the registration configuration file (see Registering with mds-servicegroup-add). The Globus Toolkit distribution includes an example configuration file: <code>\$GLOBUS_LOCATION/etc/globus_wsrfl_mds_aggregator/example-aggregator-registration.xml</code> .

The [common java client options](#)¹ are also supported.

Registering a resource manually

Prerequisites

You need the following before you register a resource with an Index Service:



Note

With GT 4.0.1, the CAS, [RFT](#)² and WS GRAM services are automatically registered with the default Index Service.

- Have a working Index Service, as documented in the [Index Service System Administrator's Guide](#)³.
- Know the EPR to the resource.
- Know the EPR to the Index Service. This can be seen in the container output at startup of the container on the index host, and will look something like this: `https://myhost:8443/wsrf/services/DefaultIndexService`

Registering with mds-servicegroup-add

To register resources:

- Create a configuration file in XML that specifies registrations. See `$GLOBUS_LOCATION/etc/globus_wsrf_mds_aggregator/example-aggregator-registration.xml` for several specific examples.
- Run `mds-servicegroup-add`⁴ to perform the registrations specified in that configuration file.

The file consists of an optional `defaultServiceGroupEPR`, an optional `defaultRegistrantEPR`, and then one or more `ServiceGroupRegistrationParameters` blocks, each of which represents one registration.

The general syntax of the configuration file is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceGroupRegistrations
  xmlns="http://mds.globus.org/servicegroup/client">

  // An optional default service group EPR.
  <defaultServiceGroupEPR>
    // Default service group EPR
  </defaultServiceGroupEPR>

  // An optional default registrant EPR.
  <defaultRegistrantEPR>
    // Default registrant EPR
  </defaultRegistrantEPR>
```

¹ <http://www.globus.org/toolkit/docs/4.0/common/javawscore/rn02re19.html>

² <http://www.globus.org/toolkit/docs/4.0/data/rft/admin-index.html#s-rft-admin-autoregistration>

³ <http://www.globus.org/toolkit/docs/4.0/info/index/>

⁴ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

```

// An optional default security descriptor file.
<defaultSecurityDescriptorFile>
  // Path name of default security descriptor file
</defaultSecurityDescriptorFile>

// One or more service group registration blocks:

<ServiceGroupRegistrationParameters>
  <ServiceGroupEPR>
    // EPR of the service group to register to
  </ServiceGroupEPR>
  <RegistrantEPR>
    // EPR of the entity to be monitored.
  </RegistrantEPR>
  <InitialTerminationTime>
    // Initial termination time
  </InitialTerminationTime>
  <RefreshIntervalSecs>
    // Refresh interval, in seconds
  </RefreshIntervalSecs>
  <Content type="agg:AggregatorContent">
    // Aggregator-source-specific configuration parameters
  </Content>
</ServiceGroupRegistrationParameters>

</ServiceGroupRegistrations>

```

The following table describes the different blocks of the file and any parameters:

Table 4. Aggregator configuration parameters

defaultService-GroupEPR block	The provides a convenient way to register a number of resources to a single service group -- for example, if you wish to register several resources to your default VO index, you can specify that index as the default service group and omit the ServiceGroupEPR blocks from each ServiceGroupRegistrationParameters block.
defaultRegistrantEPR	The provides a convenient way to register a single resource to several service groups -- for example, if you wish to register your local GRAM server to several index servers, you can specify your GRAM server as the default registrant and omit the RegistrantEPR blocks from each ServiceGroupRegistrationParameters block.
defaultSecurityDescriptorFile	Simply the path to the <u>security descriptor file</u> ⁵ .
ServiceGroupRegistrationParameters	Each ServiceGroupRegistrationParameters block specifies the parameters used to register a resource to a service group. The parameters specified in this block are:
ServiceGroupEPR	The EPR of the service group to register to. This parameter may be omitted if a defaultServiceGroupEPR block is specified; in this case, the value of defaultServiceGroupEPR will be used instead.
RegistrantEPR	The EPR of the resource to register. This parameter may be omitted if a defaultRegistrantEPR block is specified; in this case, the value of defaultRegistrantEPR will be used instead.
InitialTerminationTime	The initial termination time of this registration (this may be omitted). If the initial termination time is omitted, then the <u>mds-servicegroup-add</u> ⁶ sets the initial termination time to the current wall time plus 2 times that of the specified RefreshIntervalSecs parameter.
RefreshIntervalSecs	The refresh interval of the registration, in seconds. The <u>mds-servicegroup-add</u> ⁷ will attempt to refresh the registration according to this interval, by default incrementing the termination time of the registration by 2 times this interval for every successful refresh. If at any point the termination time for the registration expires the registration will be subject to removal within a maximum of 5 minutes.
Content	Aggregator-source-specific registration parameters. The content blocks for the various aggregator sources are described in detail in the following sections.

Simple usage for the Index Service

The simplest way to register resources to an index is to:

1. Edit the example configuration file (`$GLOBUS_LOCATION/etc/globus_wsrf_mds_aggregator/example-aggregator-registration.xml`), replacing the EPRs in that file with the EPRs for your resources
2. Run **mds-servicegroup-add** to perform the registrations specified in that file.

For example, to register to the DefaultIndexService with a modified `example-aggregator-registration.xml` file, you could run a command similar to the following:

⁵ http://www.globus.org/toolkit/docs/4.0/security/authzframe/security_descriptor.html

⁶ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

⁷ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

```
$GLOBUS_LOCATION/bin/mds-servicegroup-add -s \  
https://127.0.0.1:8443/wsrp/services/DefaultIndexService \  
$GLOBUS_LOCATION/etc/globus_wsrp_mds_aggregator/example-aggregator-registratio
```

Limitations

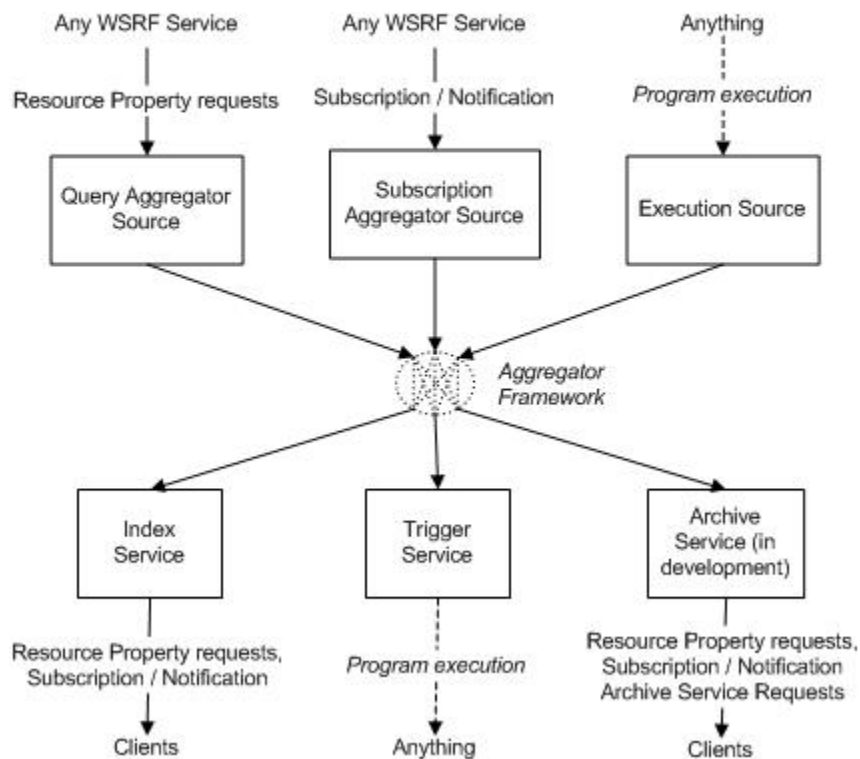
It is necessary for the tool to continue to run in order for the registrations that it maintains to be kept alive, as registrations will otherwise time out.

Chapter 7. GT 4.0 WS MDS Aggregator Framework: Aggregator Sources Reference

1. Introduction

This document describes the aggregator sources provided with the GT 4.0. *Aggregator sources* collect information from or about WS-Resources and feed that information to aggregator sinks (such as the Index Service and Trigger Service). The following graphic describes the basic information flow including the three standard aggregator sources: Query Aggregator Source, Subscription Aggregator Source and Execution Source.

Information Flow in WS-MDS



2. Types of Aggregator Sources in MDS4

The aggregator sources supplied with the toolkit collect information using resource property queries (query sources), subscription/notification (subscription sources), and execution of external programs (execution sources).

The aggregator sources supplied with the Globus Toolkit are listed in the following table.

 **Note**

All aggregator sources listed in this table are in the `org.globus.mds.aggregator.impl` package, so for example the aggregator source listed as `QueryAggregatorSource` is actually `org.globus.mds.aggregator.impl.QueryAggregatorSource`

Table 7.1. Standard aggregator sources

Aggregator Source	Description
QueryAggregatorSource	<p>The query source collects information from a registered resource by using WS-Resource Properties polling mechanisms:</p> <ul style="list-style-type: none"> • <code>GetResourcePropertyPollType</code>; requests a single Resource Property from the remote resource. • <code>GetMultipleResourcePropertiesPollType</code>; requests multiple Resource Properties from the remote resource. • <code>QueryResourcePropertiesPollType</code>; requests a query be executed against the Resource Property Set of the remote resource. <p>Polls are made periodically, with both the period and target Resource Properties specified in the registration message.</p>
SubscriptionAggregatorSource	<p>The subscription source collects information from a registered resource using WS-Notification mechanisms. Data is delivered when property values change, rather than periodically.</p>
ExecutionAggregatorSource	<p>The execution source collects information about (not necessarily from) a registered resource by execution of a local executable, which is passed as input the identity of the registered resource. Details of the interface between the execution source and local executables are in Configuring the Execution Aggregator Source¹.</p>

3. Registering Aggregator Sources

The following is general configuration information necessary for all aggregator sources (including any custom ones).

To register resources:

- Create a configuration file in XML that specifies registrations. See `$GLOBUS_LOCATION/etc/globus_ws-rf_mds_aggregator/example-aggregator-registration.xml` for several specific examples.
- Run `mds-servicegroup-add`² to perform the registrations specified in that configuration file.

The file consists of an optional `defaultServiceGroupEPR`, an optional `defaultRegistrantEPR`, and then one or more `ServiceGroupRegistrationParameters` blocks, each of which represents one registration.

The general syntax of the configuration file is:

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

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/Execution_Aggregator_Source.html

² <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

```
<ServiceGroupRegistrations
  xmlns="http://mds.globus.org/servicegroup/client">

  // An optional default service group EPR.
  <defaultServiceGroupEPR>
    // Default service group EPR
  </defaultServiceGroupEPR>

  // An optional default registrant EPR.
  <defaultRegistrantEPR>
    // Default registrant EPR
  </defaultRegistrantEPR>

  // An optional default security descriptor file.
  <defaultSecurityDescriptorFile>
    // Path name of default security descriptor file
  </defaultSecurityDescriptorFile>

  // One or more service group registration blocks:

  <ServiceGroupRegistrationParameters>
    <ServiceGroupEPR>
      // EPR of the service group to register to
    </ServiceGroupEPR>
    <RegistrantEPR>
      // EPR of the entity to be monitored.
    </RegistrantEPR>
    <InitialTerminationTime>
      // Initial termination time
    </InitialTerminationTime>
    <RefreshIntervalSecs>
      // Refresh interval, in seconds
    </RefreshIntervalSecs>
    <Content type="agg:AggregatorContent">
      // Aggregator-source-specific configuration parameters
    </Content>
  </ServiceGroupRegistrationParameters>

</ServiceGroupRegistrations>
```

The following table describes the different blocks of the file and any parameters:

Table 7.2. Aggregator configuration parameters

defaultServiceGroupEPR block	The provides a convenient way to register a number of resources to a single service group -- for example, if you wish to register several resources to your default VO index, you can specify that index as the default service group and omit the ServiceGroupEPR blocks from each ServiceGroupRegistrationParameters block.
defaultRegistrantEPR	The provides a convenient way to register a single resource to several service groups -- for example, if you wish to register your local GRAM server to several index servers, you can specify your GRAM server as the default registrant and omit the RegistrantEPR blocks from each ServiceGroupRegistrationParameters block.
defaultSecurityDescriptorFile	Simply the path to the <u>security descriptor file</u> ³ .
ServiceGroupRegistrationParameters	Each ServiceGroupRegistrationParameters block specifies the parameters used to register a resource to a service group. The parameters specified in this block are:
ServiceGroupEPR	The EPR of the service group to register to. This parameter may be omitted if a defaultServiceGroupEPR block is specified; in this case, the value of defaultServiceGroupEPR will be used instead.
RegistrantEPR	The EPR of the resource to register. This parameter may be omitted if a defaultRegistrantEPR block is specified; in this case, the value of defaultRegistrantEPR will be used instead.
InitialTerminationTime	The initial termination time of this registration (this may be omitted). If the initial termination time is omitted, then the <u>mds-servicegroup-add</u> ⁴ sets the initial termination time to the current wall time plus 2 times that of the specified RefreshIntervalSecs parameter.
RefreshIntervalSecs	The refresh interval of the registration, in seconds. The <u>mds-servicegroup-add</u> ⁵ will attempt to refresh the registration according to this interval, by default incrementing the termination time of the registration by 2 times this interval for every successful refresh. If at any point the termination time for the registration expires the registration will be subject to removal within a maximum of 5 minutes.
Content	Aggregator-source-specific registration parameters. The content blocks for the various aggregator sources are described in detail in the following sections.

4. Configuring Default Aggregator Sources

The following links provide information for configuring the three types of aggregator sources provided by the Globus Toolkit:

- [Execution Aggregator Sources Reference](#)
- [Query Aggregator Sources Reference](#)
- [Subscription Aggregator Sources Reference](#)

³ http://www.globus.org/toolkit/docs/4.0/security/authzframe/security_descriptor.html

⁴ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

⁵ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

Chapter 8. GT 4.0 WS MDS Aggregator Framework: Configuring the Execution Aggregator Source

1. Introduction

The execution aggregation source provides a way to aggregate data (arbitrary XML information) about a registered resource using an arbitrary local executable (such as an external script). The executable will be passed registration information as parameters and is expected to output the gathered data, as detailed below.

A basic example of the use of this API is described in the [ping test example for the aggregator execution source](#)¹

The execution aggregation source will periodically execute an identified executable. The identity of the executable and the frequency with which it is to run are specified in the registration message.

2. Registering

To register resources:

- Create a configuration file in XML that specifies registrations. See `$GLOBUS_LOCATION/etc/globus_ws-rf_mds_aggregator/example-aggregator-registration.xml` for several specific examples.
- Run `mds-servicegroup-add`² to perform the registrations specified in that configuration file.

The file consists of an optional `defaultServiceGroupEPR`, an optional `defaultRegistrantEPR`, and then one or more `ServiceGroupRegistrationParameters` blocks, each of which represents one registration.

The general syntax of the configuration file is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceGroupRegistrations
  xmlns="http://mds.globus.org/servicegroup/client">

  // An optional default service group EPR.
  <defaultServiceGroupEPR>
    // Default service group EPR
  </defaultServiceGroupEPR>

  // An optional default registrant EPR.
  <defaultRegistrantEPR>
    // Default registrant EPR
  </defaultRegistrantEPR>

  // An optional default security descriptor file.
```

¹ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/developer/pingtest.html>

² <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

```
<defaultSecurityDescriptorFile>
  // Path name of default security descriptor file
</defaultSecurityDescriptorFile>

// One or more service group registration blocks:

<ServiceGroupRegistrationParameters>
  <ServiceGroupEPR>
    // EPR of the service group to register to
  </ServiceGroupEPR>
  <RegistrantEPR>
    // EPR of the entity to be monitored.
  </RegistrantEPR>
  <InitialTerminationTime>
    // Initial termination time
  </InitialTerminationTime>
  <RefreshIntervalSecs>
    // Refresh interval, in seconds
  </RefreshIntervalSecs>
  <Content type="agg:AggregatorContent">
    // Aggregator-source-specific configuration parameters
  </Content>
</ServiceGroupRegistrationParameters>

</ServiceGroupRegistrations>
```

The following table describes the different blocks of the file and any parameters:

Table 8.1. Aggregator configuration parameters

defaultService- GroupEPR block	The provides a convenient way to register a number of resources to a single service group -- for example, if you wish to register several resources to your default VO index, you can specify that index as the default service group and omit the ServiceGroupEPR blocks from each ServiceGroupRegistrationParameters block.
defaultRegistrantE- PR	The provides a convenient way to register a single resource to several service groups -- for example, if you wish to register your local GRAM server to several index servers, you can specify your GRAM server as the default registrant and omit the RegistrantEPR blocks from each ServiceGroupRegistrationParameters block.
defaultSecurity- DescriptorFile	Simply the path to the <u>security descriptor file</u> ³ .
ServiceGroupRegis- trationParameters	Each ServiceGroupRegistrationParameters block specifies the parameters used to register a resource to a service group. The parameters specified in this block are:
ServiceGroupEPR	The EPR of the service group to register to. This parameter may be omitted if a defaultServiceGroupEPR block is specified; in this case, the value of defaultServiceGroupEPR will be used instead.
RegistrantEPR	The EPR of the resource to register. This parameter may be omitted if a defaultRegistrantEPR block is specified; in this case, the value of defaultRegistrantEPR will be used instead.
InitialTermination- Time	The initial termination time of this registration (this may be omitted). If the initial termination time is omitted, then the <u>mds-servicegroup-add</u> ⁴ sets the initial termination time to the current wall time plus 2 times that of the specified RefreshIntervalSecs parameter.
RefreshInter- valSecs	The refresh interval of the registration, in seconds. The <u>mds-servicegroup-add</u> ⁵ will attempt to refresh the registration according to this interval, by default incrementing the termination time of the registration by 2 times this interval for every successful refresh. If at any point the termination time for the registration expires the registration will be subject to removal within a maximum of 5 minutes.
Content	Aggregator-source-specific registration parameters. The content blocks for the various aggregator sources are described in detail in the following sections.

3. Configuration file: parameters for the execution aggregator source

The configuration block for ExecutionAggregatorSource (inside the Content block) looks like this:

```
<Content xsi:type="agg:AggregatorContent"
  xmlns:agg="http://mds.globus.org/aggregator/types">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
    <agg:ExecutionPollType>
```

³ http://www.globus.org/toolkit/docs/4.0/security/authzframe/security_descriptor.html

⁴ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

⁵ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

```
<agg:PollIntervalMillis>interval_in_ms</agg:PollIntervalMillis>
<agg:ProbeName>dummy_namespace:probe_name</agg:ProbeName>
</agg:ExecutionPollType>
</agg:AggregatorConfig>
<agg:AggregatorData/>
</Content>
```

where:

PollIntervalMillis This parameter is the poll refresh period in milliseconds.

ProbeName This parameter specifies name of the probe to run. This probe is defined in the `jndi-config.xml` file for the service being configured (for example, the file for the MDS Index service is `$GLOBUS_LOCATION/etc/globus_wsrp_mds_index_jndi-config.xml`). An `executableMappings` parameter should be defined within this file to map probe names to executable names. For example, this maps the probe names `aggr-test` and `pingexec` to the executables called `aggregator-exec-test.sh` and `example-ping-exec`, respectively. All executables are presumed to be in the directory `$GLOBUS_LOCATION/libexec/aggrexec`.

```
<resource name="configuration"
  type="org.globus.mds.aggregator.impl.AggregatorConfigurati
  <resourceParams>
    // ...
    <parameter>
      <name>executableMappings</name>
      <value>aggr-test=aggregator-exec-test.sh, pingexec=example-ping-
    </parameter>
  </resourceParams>
</resource>
```

4. Troubleshooting

If you've properly configured and registered your script for execution but are getting errors from the container because it cannot find the specified script, there are two likely causes.

First, make sure that your script/program is executable and is located in the `$GLOBUS_LOCATION/libexec/aggrexec` directory. When it's specified in the configuration mentioned above, only specify the name of the script/program, without any qualification or path. For example, using the `ProbeName` as `test-script` will be specifying the file `$GLOBUS_LOCATION/libexec/aggrexec/test-script` script.

Next, make sure that you have correctly created an `executableMappings` definition in the appropriate `jndi-config.xml` file.

5. Configuring the executable

5.1. Name of executable

The executable to run will be `$GLOBUS_LOCATION/libexec/aggrexec/<scriptname>` with `scriptname` supplied by the `ProbeName` parameter in the configuration file.

5.2. Input to executable

Information about the registration will be supplied as command line parameters and on stdin.

A single command line parameter will be supplied to the executable. This will be the URL from the EPR of the registered service.

Two XML documents will be sent to stdin, in sequence:

1. The first document will be the full EPR to the registered service.
2. The second document will be the AggregatorConfig block from the registration message (configuration file).

5.3. Output from executable

The executable must output a well-formed XML document to stdout. This output document will be delivered into the aggregator framework.

Chapter 9. GT 4.0 WS MDS Aggregator Framework: Configuring the Query Aggregator Source

1. Introduction

The QueryAggregatorSource collects information from a registered resource by using WS-Resource Properties polling mechanisms.

2. Registering

To register resources:

- Create a configuration file in XML that specifies registrations. See `$GLOBUS_LOCATION/etc/globus_ws-rf_mds_aggregator/example-aggregator-registration.xml` for several specific examples.
- Run `mds-servicegroup-add`¹ to perform the registrations specified in that configuration file.

The file consists of an optional `defaultServiceGroupEPR`, an optional `defaultRegistrantEPR`, and then one or more `ServiceGroupRegistrationParameters` blocks, each of which represents one registration.

The general syntax of the configuration file is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceGroupRegistrations
  xmlns="http://mds.globus.org/servicegroup/client">

  // An optional default service group EPR.
  <defaultServiceGroupEPR>
    // Default service group EPR
  </defaultServiceGroupEPR>

  // An optional default registrant EPR.
  <defaultRegistrantEPR>
    // Default registrant EPR
  </defaultRegistrantEPR>

  // An optional default security descriptor file.
  <defaultSecurityDescriptorFile>
    // Path name of default security descriptor file
  </defaultSecurityDescriptorFile>

  // One or more service group registration blocks:

  <ServiceGroupRegistrationParameters>
```

¹ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

```
<ServiceGroupEPR>
  // EPR of the service group to register to
</ServiceGroupEPR>
<RegistrantEPR>
  // EPR of the entity to be monitored.
</RegistrantEPR>
<InitialTerminationTime>
  // Initial termination time
</InitialTerminationTime>
<RefreshIntervalSecs>
  // Refresh interval, in seconds
</RefreshIntervalSecs>
<Content type="agg:AggregatorContent">
  // Aggregator-source-specific configuration parameters
</Content>
</ServiceGroupRegistrationParameters>

</ServiceGroupRegistrations>
```

The following table describes the different blocks of the file and any parameters:

Table 9.1. Aggregator configuration parameters

defaultService- GroupEPR block	The provides a convenient way to register a number of resources to a single service group -- for example, if you wish to register several resources to your default VO index, you can specify that index as the default service group and omit the ServiceGroupEPR blocks from each ServiceGroupRegistrationParameters block.
defaultRegistrantE- PR	The provides a convenient way to register a single resource to several service groups -- for example, if you wish to register your local GRAM server to several index servers, you can specify your GRAM server as the default registrant and omit the RegistrantEPR blocks from each ServiceGroupRegistrationParameters block.
defaultSecurity- DescriptorFile	Simply the path to the <u>security descriptor file</u> ² .
ServiceGroupRegis- trationParameters	Each ServiceGroupRegistrationParameters block specifies the parameters used to register a resource to a service group. The parameters specified in this block are:
ServiceGroupEPR	The EPR of the service group to register to. This parameter may be omitted if a defaultServiceGroupEPR block is specified; in this case, the value of defaultServiceGroupEPR will be used instead.
RegistrantEPR	The EPR of the resource to register. This parameter may be omitted if a defaultRegistrantEPR block is specified; in this case, the value of defaultRegistrantEPR will be used instead.
InitialTermination- Time	The initial termination time of this registration (this may be omitted). If the initial termination time is omitted, then the <u>mds-servicegroup-add</u> ³ sets the initial termination time to the current wall time plus 2 times that of the specified RefreshIntervalSecs parameter.
RefreshInter- valSecs	The refresh interval of the registration, in seconds. The <u>mds-servicegroup-add</u> ⁴ will attempt to refresh the registration according to this interval, by default incrementing the termination time of the registration by 2 times this interval for every successful refresh. If at any point the termination time for the registration expires the registration will be subject to removal within a maximum of 5 minutes.
Content	Aggregator-source-specific registration parameters. The content blocks for the various aggregator sources are described in detail in the following sections.

3. Configuration file: parameters for the query aggregator source

The QueryAggregatorSource can use one of the following three configuration blocks in the Content block: GetResourcePropertyPollType, GetMultipleResourcePropertiesPollType and QueryResourcePropertiesPollType.

3.1. GetResourcePropertyPollType

If a GetResourcePropertyPollType block is used, QueryAggregatorSource will request a single resource property. The block has this form:

² http://www.globus.org/toolkit/docs/4.0/security/authzframe/security_descriptor.html

³ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

⁴ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

```
<Content xsi:type="agg:AggregatorContent"
  xmlns:agg="http://mds.globus.org/aggregator/types">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
    <agg:GetResourcePropertyPollType>
      <agg:PollIntervalMillis>interval_in_ms</agg:PollIntervalMillis>
      <agg:ResourcePropertyName>rp_namespace:rp_localname</agg:ResourcePropertyName>
    </agg:GetResourcePropertyPollType>
  </agg:AggregatorConfig>
  <agg:AggregatorData/>
</Content>
```

where:

PollIntervalMillis This parameter is the poll refresh period in milliseconds; the **ResourcePropertyName** parameter is the QName of the resource property to poll for.

3.2. GetMultipleResourcePropertiesPollType

If a **GetMultipleResourcePropertiesPollType** block is used, **QueryAggregatorSource** will request one or more resource properties. The block has this form:

```
<Content
  xmlns:agg="http://mds.globus.org/aggregator/types"
  xsi:type="agg:AggregatorContent">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
    <agg:GetMultipleResourcePropertiesPollType>
      <agg:PollIntervalMillis>interval_in_ms</agg:PollIntervalMillis>
      <agg:ResourcePropertyNames>rp1_namespace:rp1_localname</agg:ResourcePropertyNames>
      <agg:ResourcePropertyNames>rp2_namespace:rp3_localname</agg:ResourcePropertyNames>
      <agg:ResourcePropertyNames>rp3_namespace:rp3_localname</agg:ResourcePropertyNames>
    </agg:GetMultipleResourcePropertiesPollType>
  </agg:AggregatorConfig>
  <agg:AggregatorData/>
</Content>
```

where:

PollIntervalMillis This parameter is the poll refresh period in milliseconds; the **ResourcePropertyNames** parameters are the QNames of the resource properties to poll for. There is no limit on the number of **ResourcePropertyNames** that may be specified.

3.3. QueryResourcePropertiesPollType

If a **QueryResourcePropertiesPollType** block is used, **QueryAggregatorSource** will request that a query be executed against the Resource Property Set of the remote resource. In the GT4 implementation of WSRF Core, the only query language that is supported is XPath. The block has this form:

```
<Content
  xmlns:agg="http://mds.globus.org/aggregator/types"
  xsi:type="agg:AggregatorContent">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
```

```
<agg:QueryResourcePropertiesPollType>
  <agg:PollIntervalMillis>interval_in_ms</agg:PollIntervalMillis>
  <agg:QueryExpressionDialect="dialect">
    Query Expression
  </agg:QueryExpression>
</agg:QueryResourcePropertiesPollType>
</agg:AggregatorConfig>
<agg:AggregatorData/>
</Content>
```

where:

PollIntervalMillis This parameter is the poll refresh period in milliseconds.

QueryExpression An `xsd:any` element; the `Dialect` attribute specifies the dialect of the query expression.

Chapter 10. GT 4.0 WS MDS Aggregator Framework: Configuring the Subscription Aggregator Source

1. Introduction

The SubscriptionAggregatorSource gathers resource property values from the registered resource using WS-Notification subscriptions.

2. Registering

To register resources:

- Create a configuration file in XML that specifies registrations. See `$GLOBUS_LOCATION/etc/globus_ws-rf_mds_aggregator/example-aggregator-registration.xml` for several specific examples.
- Run `mds-servicegroup-add`¹ to perform the registrations specified in that configuration file.

The file consists of an optional defaultServiceGroupEPR, an optional defaultRegistrantEPR, and then one or more ServiceGroupRegistrationParameters blocks, each of which represents one registration.

The general syntax of the configuration file is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceGroupRegistrations
  xmlns="http://mds.globus.org/servicegroup/client">

  // An optional default service group EPR.
  <defaultServiceGroupEPR>
    // Default service group EPR
  </defaultServiceGroupEPR>

  // An optional default registrant EPR.
  <defaultRegistrantEPR>
    // Default registrant EPR
  </defaultRegistrantEPR>

  // An optional default security descriptor file.
  <defaultSecurityDescriptorFile>
    // Path name of default security descriptor file
  </defaultSecurityDescriptorFile>

  // One or more service group registration blocks:

  <ServiceGroupRegistrationParameters>
```

¹ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

GT 4.0 WS MDS Aggregator Framework: Configuring the Subscription Aggregator Source

```
<ServiceGroupEPR>
  // EPR of the service group to register to
</ServiceGroupEPR>
<RegistrantEPR>
  // EPR of the entity to be monitored.
</RegistrantEPR>
<InitialTerminationTime>
  // Initial termination time
</InitialTerminationTime>
<RefreshIntervalSecs>
  // Refresh interval, in seconds
</RefreshIntervalSecs>
<Content type="agg:AggregatorContent">
  // Aggregator-source-specific configuration parameters
</Content>
</ServiceGroupRegistrationParameters>

</ServiceGroupRegistrations>
```

The following table describes the different blocks of the file and any parameters:

Table 10.1. Aggregator configuration parameters

defaultService-GroupEPR block	The provides a convenient way to register a number of resources to a single service group -- for example, if you wish to register several resources to your default VO index, you can specify that index as the default service group and omit the ServiceGroupEPR blocks from each ServiceGroupRegistrationParameters block.
defaultRegistrantEPR	The provides a convenient way to register a single resource to several service groups -- for example, if you wish to register your local GRAM server to several index servers, you can specify your GRAM server as the default registrant and omit the RegistrantEPR blocks from each ServiceGroupRegistrationParameters block.
defaultSecurityDescriptorFile	Simply the path to the <u>security descriptor file</u> ² .
ServiceGroupRegistrationParameters	Each ServiceGroupRegistrationParameters block specifies the parameters used to register a resource to a service group. The parameters specified in this block are:
ServiceGroupEPR	The EPR of the service group to register to. This parameter may be omitted if a defaultServiceGroupEPR block is specified; in this case, the value of defaultServiceGroupEPR will be used instead.
RegistrantEPR	The EPR of the resource to register. This parameter may be omitted if a defaultRegistrantEPR block is specified; in this case, the value of defaultRegistrantEPR will be used instead.
InitialTerminationTime	The initial termination time of this registration (this may be omitted). If the initial termination time is omitted, then the <u>mds-servicegroup-add</u> ³ sets the initial termination time to the current wall time plus 2 times that of the specified RefreshIntervalSecs parameter.
RefreshIntervalSecs	The refresh interval of the registration, in seconds. The <u>mds-servicegroup-add</u> ⁴ will attempt to refresh the registration according to this interval, by default incrementing the termination time of the registration by 2 times this interval for every successful refresh. If at any point the termination time for the registration expires the registration will be subject to removal within a maximum of 5 minutes.
Content	Aggregator-source-specific registration parameters. The content blocks for the various aggregator sources are described in detail in the following sections.

3. Configuration file: parameters for the subscription aggregator source

The configuration block for SubscriptionAggregatorSource looks like this:

```
<Content
  xmlns:agg="http://mds.globus.org/aggregator/types"
  xsi:type="agg:AggregatorContent">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
```

² http://www.globus.org/toolkit/docs/4.0/security/authzframe/security_descriptor.html

³ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

⁴ <http://www.globus.org/toolkit/docs/4.0/info/aggregator/re01.html>

GT 4.0 WS MDS Aggregator Framework: Configuring the Subscription Aggregator Source

```
<agg:AggregatorSubscriptionType>
  <TopicExpression Dialect="dialect">
    Topic Expression
  </TopicExpression>
  <Precondition Dialect="dialect">
    Precondition
  </Precondition>
  <Selector Dialect="dialect">
    Selector
  </Selector>
  <SubscriptionPolicy>
    Subscription Policy
  </SubscriptionPolicy>
  <InitialTerminationTime>time</InitialTerminationTime>
</agg:AggregatorSubscriptionType>
</agg:AggregatorConfig>
<agg:AggregatorData/>
</Content>
```

where:

TopicExpression This is the only required parameter; it specifies the topic expression to use in the subscription request.

[TODO: link to core notification/subscription docs].

Chapter 11. GT 4.0.8 Incremental Release Notes: WS MDS Aggregator Framework

1. Introduction

These release notes are for the incremental release 4.0.8. It includes a summary of changes since 4.0.7, bug fixes since 4.0.7 and any known problems that still exist at the time of the 4.0.8 release. This page is in addition to the top-level 4.0.8 release notes at <http://www.globus.org/toolkit/releasenotes/4.0.8>.

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [WS MDS Aggregator 4.0 Release Notes](#)¹.

2. Changes Summary

No changes have been made since the previous release.

3. Bug Fixes

No bugs have been fixed in the Aggregator framework since the previous release.

4. Known Problems

There are no known problems with the WS MDS Aggregator API at the time of this release.

5. For More Information

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

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Release_Notes.html

² [index.html](#)

Chapter 12. GT 4.0.7 Incremental Release Notes: WS MDS Aggregator Framework

1. Introduction

These release notes are for the incremental release 4.0.7. It includes a summary of changes since 4.0.6, bug fixes since 4.0.6 and any known problems that still exist at the time of the 4.0.7 release. This page is in addition to the top-level 4.0.7 release notes at <http://www.globus.org/toolkit/releasenotes/4.0.7>.

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [WS MDS Aggregator 4.0 Release Notes](#)¹.

2. Changes Summary

No changes have been made since the previous release.

3. Bug Fixes

No bugs have been fixed in the Aggregator framework since the previous release.

4. Known Problems

There are no known problems with the WS MDS Aggregator API at the time of this release.

5. For More Information

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

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Release_Notes.html

² [index.html](#)

Chapter 13. GT 4.0.6 Incremental Release Notes: WS MDS Aggregator Framework

1. Introduction

These release notes are for the incremental release 4.0.6. It includes a summary of changes since 4.0.5, bug fixes since 4.0.5 and any known problems that still exist at the time of the 4.0.6 release. This page is in addition to the top-level 4.0.6 release notes at <http://www.globus.org/toolkit/releasenotes/4.0.6>.

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [WS MDS Aggregator 4.0 Release Notes](#)¹.

2. Changes Summary

No changes have been made since the previous release.

3. Bug Fixes

No bugs have been fixed in the Aggregator framework since 4.0.5.

4. Known Problems

There are no known problems with the WS MDS Aggregator API at the time of the 4.0.6 release.

5. For More Information

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

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Release_Notes.html

² [index.html](#)

Chapter 14. GT 4.0.5 Incremental Release Notes: WS MDS Aggregator Framework

1. Introduction

These release notes are for the incremental release 4.0.5. It includes a summary of changes since 4.0.4, bug fixes since 4.0.4 and any known problems that still exist at the time of the 4.0.5 release. This page is in addition to the top-level 4.0.5 release notes at <http://www.globus.org/toolkit/releasenotes/4.0.5>.

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [WS MDS Aggregator 4.0 Release Notes](#)¹.

2. Changes Summary

Usage statistics (service name, current and total registrations, and start time) are now being collected for aggregator services.

3. Bug Fixes

No bugs have been fixed in the Aggregator framework since 4.0.4.

4. Known Problems

There are no known problems with the WS MDS Aggregator API at the time of the 4.0.5 release.

5. For More Information

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

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Release_Notes.html

² [index.html](#)

Chapter 15. GT 4.0.4 Incremental Release Notes: WS MDS Aggregator Framework

1. Introduction

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

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [WS MDS Aggregator 4.0 Release Notes](#)¹.

2. Changes Summary

No changes have been made to the WS MDS Aggregator component since 4.0.2.

3. Bug Fixes

- [Bug 4803](#):² MDS stub classes are replicated in a number of jar files

4. Known Problems

There are no known problems with the WS MDS Aggregator API at the time of the 4.0.4 release.

5. For More Information

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

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Release_Notes.html

² http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4803

³ [index.html](#)

Chapter 16. GT 4.0.3 Incremental Release Notes: WS MDS Aggregator Framework

1. Introduction

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

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [WS MDS Aggregator 4.0 Release Notes](#)¹.

2. Changes Summary

Except for a bug fix, no changes have been made to the WS MDS Aggregator component since 4.0.2.

3. Bug Fixes

The following bugs were fixed for WS MDS Aggregator since GT 4.0.2:

- [Bug 4655](#):² EntrySweeper should be a daemon thread

4. Known Problems

There are no known problems with the WS MDS Aggregator API at the time of the 4.0.3 release.

5. For More Information

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

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Release_Notes.html

² http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4655

³ [index.html](#)

Chapter 17. GT 4.0.2 Incremental Release Notes: WS MDS Aggregator Framework

1. Introduction

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

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [WS MDS Aggregator 4.0 Release Notes](#)¹.

2. Changes Summary

Except for bug fixes, no changes have been made to the WS MDS Aggregator component since 4.0.1.

3. Bug Fixes

The following bugs were fixed for WS MDS ServiceGroup:

- [Bug 3673](#):² IndexOutOfBoundsException in EntryResourceProperty

The following bugs were fixed for WS MDS Aggregator:

- [Bug 4326](#):³ ExecutionAggregatorSource input stream bugs

4. Known Problems

There are no known problems with the WS MDS Aggregator API at the time of the 4.0.2 release.

5. For More Information

Click [here](#)⁴ for more information about this component.

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Release_Notes.html

² http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=3673

³ http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4326

⁴ [index.html](#)

Chapter 18. GT 4.0.1 Incremental Release Notes: WS MDS Aggregator Framework

1. Introduction

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

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the [WS MDS Aggregator 4.0 Release Notes](#)¹.

2. Changes Summary

The following features have been added to the WS MDS Aggregator component:

- Two additional configuration options have been added to the AggregatorConfiguration JNDI bean type that allow administrators more control over when notifications are sent out to listeners that have subscribed to the service group 'Entry' resource property. These changes consist of two boolean parameters, "notifyOnEntryLifetimeExtension" and "notifyOnEntryContentChange". By default, if the parameters are not supplied, "notifyOnEntryLifetimeExtension" is set to TRUE and "notifyOnEntryContentChange" is set to FALSE. The "notifyOnEntryLifetimeExtension" parameter causes a notification to be generated whenever a service group entry is added, removed, or refreshed. The "notifyOnEntryContentChange" causes a notification to be generated whenever an entry has is aggregate content updated. The syntax for specifying these configuration parameters is as follows:

```
<resource name="configuration"
          type="org.globus.mds.aggregator.impl.AggregatorConfiguration">
  ...

  <parameter>
<name>notifyOnEntryLifetimeExtension</name>
<value>>true</value>
  </parameter>
  <parameter>
<name>notifyOnEntryContentChange</name>
<value>>false</value>
  </parameter>

  ...
</resource>
```

¹ http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Release_Notes.html

- An additional configuration option has been added to the `AggregatorConfiguration` JNDI bean type to allow administrators to configure the interval in milliseconds at which the the service group's sweeper thread purges out stale registrant entries. The syntax for specifying the sweeper interval is:

```
<resource name="configuration"
          type="org.globus.mds.aggregator.impl.AggregatorConfiguration">
  ...

  <parameter>
    <name>entrySweeperInterval</name>
    <value>120000</value>
  </parameter>

  ...
</resource>
```

- An additional configuration option has been added to the `AggregatorConfiguration` JNDI bean type to allow administrators to specify a developer supplied callback function that will be invoked when a service group entry is removed from the service group. See the API documentation for the `AggregatorServiceGroupEntryRemovedCallback` interface for implementation details.

```
<resource name="configuration"
          type="org.globus.mds.aggregator.impl.AggregatorConfiguration">
  ...

  <parameter>
    <name>entryRemovedCallback</name>
    <value>org.globus.mds.aggregator.impl.SampleAggregatorRemoveCallback</value>
  </parameter>

  ...
</resource>
```

3. Bug Fixes

The following bugs were fixed for WS MDS ServiceGroup:

- [Bug 3500](http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=3500):² `ServiceGroupRegistrationClient.terminate()` failure
- [Bug 3470](http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=3470):³ Unregister functionality missing in `ServiceGroupRegistrationClient`?

The following bugs were fixed for WS MDS Aggregator:

- [Bug 3393](http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=3393):⁴ Notifications need to be sent by an `AggregatingServiceGroupResource` when registrants successfully update the `terminationTime` of a member `AggregatingServiceGroupEntryResource`

² http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=3500

³ http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=3470

⁴ http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=3393

- [Bug 3177](#):⁵ AggregatingServiceGroup EntrySweeper interval needs to be user configurable
- [Bug 3150](#):⁶ AggregatingServiceGroup* needs to provide Resource.RemoveCallback for entry resources (or equivalent functionality) for derived code

4. Known Problems

There are no known problems with the WS MDS Aggregator API at the time of the 4.0.1 release.

5. For More Information

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

⁵ http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=3177

⁶ http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=3150

⁷ [index.html](#)

Chapter 19. GT 4.0 Release Notes: WS MDS Aggregator Framework

1. Component Overview

The Aggregator Framework is the software framework on which WS MDS services (currently, the WS MDS *Index* and WS MDS *Trigger* services) are built. The aggregator framework collects data from an *aggregator source* and sends that data to an *aggregator sink* for processing. Aggregator sources distributed with the Globus Toolkit include modules that query service data, acquire data through subscription/notification, and execute programs to generate data. Aggregator sinks include modules that implement the WS MDS Index service interface and the WS MDS Trigger service interface.

2. Feature Summary

Features new in release GT 4.0

- Ported to use WSRF mechanisms (previously used OGSF).
- Additional sources which collect information by polling and by execution of local scripts.
- Management of aggregations is now performed over the wire through WS ServiceGroup APIs.

Other Supported Features

- Collects information from grid resources using pluggable aggregation sources.
- Delivers collected information to pluggable sinks.
- Manages creation and destruction of individual aggregation registrations.

3. Bug Fixes

- [Bug 2432: Aggregator sample config file should have tls-style urls](#)¹
- [Bug 2388: rejected add\(\) leaves incomplete state behind which then upsets the sweeper](#)²
- [Bug 2384: remove stack trace from execution source error](#)³
- [Bug 2377: support multiple aggregator sources in one aggregator](#)⁴
- [Bug 2157: Aggregator entries do not implement WS-Resource Lifetime resource properties](#)⁵
- [Bug 2145: query source waits entire reg period before making first request](#)⁶

¹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2432

² http://bugzilla.globus.org/globus/show_bug.cgi?id=2388

³ http://bugzilla.globus.org/globus/show_bug.cgi?id=2384

⁴ http://bugzilla.globus.org/globus/show_bug.cgi?id=2377

⁵ http://bugzilla.globus.org/globus/show_bug.cgi?id=2157

⁶ http://bugzilla.globus.org/globus/show_bug.cgi?id=2145

- [Bug 2105: flatten schema in CVS](#)⁷
- [Bug 2097: IndexDefaultWidgetTest registers same widget many times](#)⁸
- [Bug 2087: getresourceproperty poll source does not tolerate whitespace in RP name](#)⁹
- [2082: index and aggregator stub generation is generating metrics stubs](#)¹⁰
- [2191: AggregatingServiceGroupEntries do not implement Service Group EPR properly](#)¹¹
- [All fixed aggregator framework bugs and enhancement requests](#)¹²
- [2339: Registered services connection refused messages](#)¹³
- [2398: index service does not appear to have wsrf-query support \(any more\)](#)¹⁴
- [2850: MDS4 logs to INFO when DEBUG should be used](#)¹⁵
- [2958: WS-MDS Aggregator Junit test execution has no results](#)¹⁶
- [2963: need to create libexec/aggrexec dir as part of MDS install](#)¹⁷

4. Known Problems

- [2082: index and aggregator stub generation is generating metrics stubs](#)¹⁸
- [2191: AggregatingServiceGroupEntries do not implement Service Group EPR properly](#)¹⁹
- [2339: Registered services connection refused messages](#)²⁰
- [2398: index service does not appear to have wsrf-query support \(any more\)](#)²¹
- [2807: execution aggregator source junit test needs some work](#)²²
- [All open aggregator bug reports and enhancement requests](#)²³

⁷ http://bugzilla.globus.org/globus/show_bug.cgi?id=2105

⁸ http://bugzilla.globus.org/globus/show_bug.cgi?id=2097

⁹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2087

¹⁰ http://bugzilla.globus.org/globus/show_bug.cgi?id=2082

¹¹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2191

¹² <http://bugzilla.globus.org/globus/buglist.cgi?regetlastlist=1>

¹³ http://bugzilla.globus.org/globus/show_bug.cgi?id=2339

¹⁴ http://bugzilla.globus.org/globus/show_bug.cgi?id=2398

¹⁵ http://bugzilla.globus.org/globus/show_bug.cgi?id=2850

¹⁶ http://bugzilla.globus.org/globus/show_bug.cgi?id=2958

¹⁷ http://bugzilla.globus.org/globus/show_bug.cgi?id=2963

¹⁸ http://bugzilla.globus.org/globus/show_bug.cgi?id=2082

¹⁹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2191

²⁰ http://bugzilla.globus.org/globus/show_bug.cgi?id=2339

²¹ http://bugzilla.globus.org/globus/show_bug.cgi?id=2398

²² http://bugzilla.globus.org/globus/show_bug.cgi?id=2807

²³ http://bugzilla.globus.org/globus/buglist.cgi?short_desc_type=allwordssubstr&short_desc=&product=MDS&component=wsrf_aggregator&long_desc_type=allwordssubstr&long_desc=&bug_file_loc_type=allwordssubstr&bug_file_loc=&bug_status=NEW&bug_status=AS-SIGNED&bug_status=REOPENED&emailtype1=substring&email1=&emailtype2=substring&email2=&bugidtype=include&bug_id=&votes=&channelin=&chfieldfrom=&chfieldto=Now&cmdtype=doit&newqueryname=&order=Reuse+same+sort+as+last+time&field0-0-0=noop&type0-0-0=noop&value0-0-0=

5. Technology Dependencies

Aggregator Framework depends on the following GT components:

- Java WS Core

Aggregator Framework depends on the following 3rd party software:

- None

6. Tested Platforms

Tested Platforms for WS MDS Aggregator Framework

- Linux on i386
- Windows XP

7. Backward Compatibility Summary

Protocol changes since GT version 3.2

- The aggregator framework is a complete reimplementaion of the MDS3 aggregator framework using WSRF rather than OGSi protocols.
- No wireside compatibility with MDS3 aggregator framework.
- Architectural similarity should make porting straightforward.

API changes since GT version 3.2

- APIs entirely rewritten, so no API compatibility.
- Architectural similarity should make porting straightforward.

Exception changes since GT version 3.2

- See API changes above.

Schema changes since GT version 3.2

- Registration interface uses WSRF rather than OGSi schemas.
- New per-source and per-sink configuration schemas.

8. For More Information

Click [here](#)²⁴ for more information about this component.

²⁴ index.html

GT 4.0 WS MDS Glossary

A

Aggregator Framework	A software framework used to build services that collect and aggregate data. MDS4 Services (such as the Index and Trigger services) are built on the Aggregator Framework, and are sometimes called Aggregator Services.
aggregator services	Services that are built on the Aggregator Framework, such as the MDS4 Index Service and Trigger Service. See Also Aggregator Framework , Index Service , Trigger Service .
aggregator source	A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. MDS4 contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source. See Also query aggregator source , subscription aggregator source , execution aggregator source .

E

execution aggregator source	An Aggregator Source (included in MDS4) that executes an administrator-supplied program to collect information and make it available to an Aggregator Service such as the Index Service. See Also aggregator source .
-----------------------------	--

G

Ganglia	A cluster monitoring tool. See http://ganglia.sourceforge.net .
---------	--

H

Hawkeye	A monitoring service for Condor Pools. See http://www.cs.wisc.edu/condor/hawkeye/ .
---------	--

I

Index Service	An aggregator service that serves as a registry similar to UDDI, but much more flexible. Indexes collect information and publish that information as WSRF resource properties. See Also aggregator services .
information provider	A "helper" software component that collects or formats resource information, for use by an aggregator source or by a WSRF service when creating resource properties.

Q

query aggregator source An aggregator source (included in MDS4) that polls a WSRF service for resource property information.
See Also [aggregator source](#).

S

subscription aggregator source An aggregator source (included in MDS4) that collects data from a WSRF service via WSRF subscription/notification.
See Also [aggregator source](#).

T

Trigger Service An aggregator service that collects information and compares that data against a set of conditions defined in a configuration file. When a condition is met, or triggered, an action takes place, such as emailing a system administrator when the disk space on a server reaches a threshold.
See Also [aggregator services](#).

W

WebMDS A web-based interface to WS-RF resource property information that can be used as a user-friendly front-end to the Index Service or other WS-RF services.