GT 4.2.1 WS MDS Trigger Service: Public Interface Guide

GT 4.2.1 WS MDS Trigger Service: Public Interface Guide

Table of Contents

List of Tables

Chapter 1. APIs

1. Programming Model Overview

Information about how to configure existing aggregator sources (such as the aggregator sources distributed with the Globus Toolkit, which include one that polls for resource property information, one that collects resource property information through subscription/notification, and one that collects information by executing an executable program) is found in Aggregator Sources Reference; information about how to create new aggregator sources can be found in Developer's Guide.

The administrator of a Globus installation configures the set of available executable programs that are available to be used as action scripts (for example, an executable program may send mail to an end-user or write a structured log file that will later be read by some other program).

Chapter 2. WS and WSDL

1. Protocol overview

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

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

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

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

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

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

2. Operations

2.1. AggregatorServiceGroup

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

2.2. AggregatorServiceGroupEntry

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

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

3. WS MDS Aggregator Framework Resource Properties

3.1. AggregatorServiceGroup Resource Properties

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

4. Faults

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

5. WSDL and Schema Definition

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

Other relevant source files are the:

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

[3] http://viewcvs.globus.org/viewcvs.cgi/ws-mds/aggregator/schema/mds/aggregator/aggregator_service_group_port_type.wsdl?revision=1.5&view=markup&pathrev=globus_4_2_branch
[4] http://viewcvs.globus.org/viewcvs.cgi/ws-mds/aggregator/schema/mds/aggregator/aggregator_service_group_entry_port_type.wsdl?revision=1.6&view=markup&pathrev=globus_4_2_branch
[5] http://viewcvs.globus.org/viewcvs.cgi/ws-mds/aggregator/schema/mds/aggregator/aggregator-types.xsd?revision=1.6&view=markup&pathrev=globus_4_2_branch
[6] http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/servicegroup/sgw-2.wsdl?revision=1.2&view=markup&pathrev=globus_4_2_branch
[7] http://viewcvs.globus.org/viewcvs.cgi/wsrf/schema/wsrf/lifetime/rlw-2.wsdl?revision=1.2&view=markup&pathrev=globus_4_2_branch

Chapter 3. Additional WSDL information for the Trigger Service

1. Trigger Service Resource Properties

In addition to the resource properties for the Aggregator Framework, the Trigger Service exposes the following:

`TriggerName` This resource property allows one to arbitrarily name the trigger. This is used to assist one in managing many triggers.

`TriggerStatus` This resource property is used to indicate the current status of the trigger. There are two states allowed: `enabled` and `disabled`.

`MemberEPR` This resource property reveals the monitored service that the trigger is associated with

`TriggerID` This resource property is a unique ID assigned to the trigger. It is essentially the EPR's Resource Key.

`MatchingRule` This resource property is the XPath expression that will be used in evaluating incoming aggregator data. The trigger will fire (if enabled) if the expression is "true" (in a boolean sense). But if "EnableBoolean" is set to "false", then if the MatchingRule returns any data, the trigger will fire. This is consistent with pre-4.2 trigger functionality.

`NamespaceMappings` This resource property allows one to use namespaces in the MatchingRule.

`ActionScript` This resource property is the name of the action script that should be fired when the trigger evaluation is "true". The action script is located in the `$GLOBUS_LOCATION/libexec/trigger/` directory.

`EnableBoolean` This resource property is by default `true`, meaning that it is set up to evaluate XPath queries as "true" or "false", firing only when "true". If this property is set to "false", then the trigger will fire only if the MatchingRule evaluation returns any data

`MinimumFiringInterval` The action script will not be executed more than once in this number of seconds. If unspecified, there will be no minimum interval.

`MinimumMatchTime` The MatchingRule must be true for this number of seconds before the ActionScript will be executed. If unspecified, there is no minimum time period that the rule must match and the rule will be eligible to fire immediately after the MatchingRule becomes true.

`StartTime` The trigger will not fire, nor will the TriggerService perform any evaluations before the StartTime, if indicated. If a start time is not indicated, the TriggerService will begin immediately performing evaluations, if the trigger is active (i.e. TriggerStatus is set to "enabled")

`EndTime` The TriggerService will not perform any evaluations after the EndTime, if indicated. If an end time is not indicated, the TriggerService will continue performing evaluations (of "active" triggers) until an EndTime is specified, otherwise until

	the container is shutdown. After an EndTime has passed, the TriggerService will basically be doing nothing, but you may whenever you wish change the EndTime, and the trigger evaluations will then begin again until the EndTime again is reached.
InvalidityStartTime	The trigger will not fire, nor will the TriggerService perform any evaluations after the InvalidityStartTime, if indicated. If an InvalidityStartTime is indicated, an InvalidityEndTime must also be specified. During this time period, the TriggerService will not perform any evaluations, if the trigger is active (i.e. TriggerStatus is set to "enabled")
InvalidityEndTime	This parameter requires an InvalidityStartTime, and during the time period between the InvalidityStartTime and InvalidityEndTime, the TriggerService will not perform any evaluations. If there is an EndTime specified, then trigger evaluations will begin after the InvalidityEndTime until the EndTime.
ActionScriptInputFullOriginal	This parameter, if set to "true" will pass the original trigger message input (to which the matching rule was applied) to the action script. The default behavior is to always pass the entire input message to the action scripts. For action scripts which do not need to consume the unmodified input, this variable may be set to "false" in order to increase performance. For users familiar with previous versions of the Trigger Service, if you set ActionScriptInputFullOriginal to "true", this is equivalent to setting disableUnmodifiedActionScriptInput to "false", in other words it will pass the original trigger message input (to which the matching rule was applied) to the action script.
ActionScriptInputXPathQueryResult	If this value is present and set to true, the service will pass the filtered output result of the XPath MatchingRule as additional input to the stdin of the action script, in addition to the original input to which the matching rule was applied. The default behavior if unspecified is true, meaning the Xpath query result will be passed as input to the action script. For users familiar with previous versions of the Trigger Service, if you set ActionScriptInputXpathQueryResult to "true", this is equivalent to setting enableFilteredActionScriptInput to "true".

The following resource properties are not editable; they are trigger run-time statistics.

RuleLastCheckedAt	This resource property reveals when the MatchingRule was last checked/evaluated.
ConditionTrueSince	This resource property reports since when the MatchingRule evaluated against the incoming aggregator data results in `true`
ActionFiredAt	This resource property reveals exactly when the trigger was last fired.

Chapter 4. MDS Trigger Commands

The mds-servicegroup-add(1) command in the Aggregator Framework is used to configure the Trigger Registration Service to collect information from various sources. In addition, the Trigger Service has three command-line clients

1. Create a new trigger - mds-trigger-create

Synopsis

```
mds-trigger-create [options] -b baseURL monitoredURL
```

Description

This command creates a new trigger.

Table 4.1. mds-trigger-create options

-b baseURL	Specify the trigger service's base URL (everything in the Trigger Service URL up to the service name). This option is used instead of the customary `-s` or `-e` options because this client communicates with more than one trigger-related service.
monitoredURL	Specify the URL of the service to be monitored; this should be the same as the address of a service registered to the Trigger Registry Service's service group.

Example

The first command creates a new trigger on the server *triggerhost.org* to monitor the information in the default Index Server running in the same Globus container. The second command creates a new trigger on the server *triggerhost.org* to monitor the information in an Index Server running on the server *otherhost.org*

```
mds-trigger-create -b https://triggerhost.org:8443/wsrf/services \
   https://triggerhost.org:8443/wsrf/services/DefaultIndexService

mds-trigger-create -b https://triggerhost.org:8443/wsrf/services \
   https://otherhost.org:8443/wsrf/services/DefaultIndexService
```

2. View information about existing trggers - mds-trigger-view

Synopsis

```
mds-trigger-view [options] -b baseURL [TriggerID]
```

Description

This displays information about triggers.

Table 4.2. mds-trigger-view options

-b baseURL	Specify the trigger service's base URL (everything in the Trigger Service URL up to the service name). This option is used instead of the customary `-s` or `-e` options because this client communicates with more than one trigger-related service.
TriggerID	If a Trigger ID is specified, detailed information about the specified trigger will be displayed; if not, summary information about all triggers will be displayed.

Example

The first command displays summary information about all triggers known to the Trigger Service; the second displays detailed information about one trigger

```
mds-trigger-view -b https://triggerhost.org:8443/wsrf/services

mds-trigger-view -b https://triggerhost.org:8443/wsrf/services \
   546aae00-418b-11dd-a5ea-ebfac2dfbbee
```

3. Modify a trgger - mds-trigger-edit

Synopsis

```
mds-trigger-edit [options] -b baseURL TriggerID Parameter=Value
```

Description

This command is used to modify trigger parameters, in order to change the trigger conditions, actions, status (enabled or disabled), etc.

Table 4.3. mds-trigger-edit options

-b baseURL	Specify the trigger service's base URL (everything in the Trigger Service URL up to the service name). This option is used instead of the customary `-s` or `-e` options because this client communicates with more than one trigger-related service.
TriggerID	The identifier of the trigger to be modified
Param=value	Set the named parameter to the specified value. The parameter can be any writable Trigger Service resource property

Examples

The first command enables a trigger; the second command disables it.

```
mds-trigger-edit -b https://triggerhost.org:8443/wsrf/services \
   546aae00-418b-11dd-a5ea-ebfac2dfbbee \
   TriggerStatus=enabled

mds-trigger-edit -b https://triggerhost.org:8443/wsrf/services \
   546aae00-418b-11dd-a5ea-ebfac2dfbbee \
   TriggerStatus=disabled
```

Change the trigger condition (matching rule) so that the trigger fires if there are no Index Service entries.

```
mds-trigger-edit -b https://triggerhost.org:8443/wsrf/services \
   546aae00-418b-11dd-a5ea-ebfac2dfbbee \
   MatchingRule="count(//*[local-name()='Entry'])=0"
```

Change the trigger action.

```
mds-trigger-edit -b https://triggerhost.org:8443/wsrf/services \
   546aae00-418b-11dd-a5ea-ebfac2dfbbee \
   ActionScript=trigger-action-input-default
```

Chapter 5. Additional configuration for the Trigger Service

1. Additional configuration for the Trigger Service

The set of possible actions (programs) that can be executed by the Trigger Service is specified in the file `$GLOBUS_LOCATION/etc/globus_wsrf_mds_trigger/jndi-config.xml`. The `executableMappings` parameter contains a comma-separated list of name=value pairs. The left hand side of each name/value pair is the name assigned to the trigger action; this name can be used in trigger definitions. The right hand side of each name/value pair is the path name (relative to `$GLOBUS_LOCATION/libexec/trigger` of the file to execute.

The sources of information used by the Trigger Service are configured using the `mds-servicegroup-add` command; see the Aggregator Framework documentation or the Trigger Service Easy HOWTO for more details and examples.

Triggers themselves are created using the command line clients.

Chapter 6. Graphical User Interface

The release contains WebMDS which can be used to display the status of resources registered to a Trigger Service in a normal web browser.

Chapter 7. Trigger Action Script

1. Format of action script stdout

The action script should output an XML document to stdout. The xml document does not need to match any particular schema. This output will be included in the ServiceGroupEntry for the rule.

Chapter 8. Aggregator sources

The public interfaces for creating and configuring aggregator sources -- sources of information used by the trigger service -- can be found in Aggregator Sources Reference.

Appendix A. Errors

Table A.1. WS MDS Trigger Service Error Messages

Error Code	Definition	Possible Solutions
`Error ; nested exception is: org.apache.commons.httpclient. NoHttpResponseException: The server xxx.x.x.x failed to respond`	Happens when trying to create a trigger for the Trigger Service. The above error is accompanied by the following error in container: `[JWSCORE-192] Error processing request java.io.IOException: Token length 1347375956 > 33554432`. FIXME - what causes this?	Be sure that you have properly edited the `client-config-settings` file under `globus_wsrf_mds_trigger`. The `DefaultServiceAddress` parameter should properly reflect the service prefix from your container, e.g.: `https://127.0.0.1:8444/wsrf/services/`. The services you wish to monitor should also be consistent.
WS MDS is built on Java WS Core, please see Java WS Core Error Codes for more error code documentation.		

Glossary