GT 4.2.0 WS MDS WebMDS: Developer's Guide

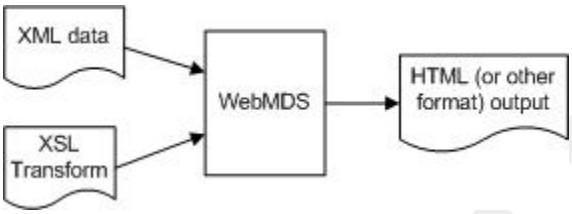
DRAFT

GT 4.2.0 WS MDS WebMDS: Developer's Guide

Introduction

WebMDS is a web-based interface for viewing formatted information about Grid resources. Information is collected via a plugin interface and then formatted using an XSLT transform.

Figure 1. WebMDS Information Flow



DRAFT

Table of Contents

DRAFT

List of Figures

DRAFT

List of Tables

DRAFT

WebMDS Howtos

Symbols

A

C

D

E

F

G

I

P

S

T

U

W

X

DRAFT

Chapter 1. Before you begin

1. Feature summary

Features new in release 4.2.0:

- None

Other Supported Features

- Extensible plugin interface to support various mechanisms to gather monitoring information and XSLT transforms.
- Plugins to acquire monitoring information via resource property mechanisms.
- Plugin to acquire XSLT transforms by reading from local files.

Deprecated Features

- None

2. Tested platforms

Tested Platforms for WebMDS:

- The WebMDS server has only been tested with Tomcat version 5.0.28; it has been tested on RedHat Linux (i386) and, to a lesser extent, on Windows XP.
- On the client side, WebMDS should be accessible from any web browser on any platform.

2.1. Installing WebMDS on Windows

Although the WebMDS server is not officially supported on non-Unix platforms, and no Windows installer exists for WebMDS, it is possible to run WebMDS on Windows. The following instructions describe how to install WebMDS on a Windows platform.

1. Install Tomcat[1] and set your `CATALINA_HOME` environment variable to the directory into which Tomcat was installed.
2. Install the Globus Java WS-Core distribution from the Globus Toolkit download page[2]. Set your `GLOBUS_LOCATION` environment variable to the directory into which you installed Globus Java WS-Core
3. Check the `ws-mds` distribution out of the Globus CVS repository[3], using the `globus_4_0_branch` tag.
4. Install the servicegroup package:

```
cd c:\wherever\ws-mds\servicegroup\schema
ant deploy
```

[1] http://jakarta.apache.org/tomcat/
[2] http://www.globus.org/toolkit/downloads/
[3] http://www.globus.org/toolkit/docs/development/remote-cvs.html

```
cd ..\source
ant deploy
```

where *wherever* is the directory into which you checked out the ws-mds sources.

5. Install WebMDS:

```
cd c:\wherever\ws-mds\webmds
ant deploy
```

6. Create the webmds context file (this tells Tomcat where to find WebMDS):

```
%GLOBUS_LOCATION%\lib\webmds\bin\webmds-create-context-file %CATALINA_HOME%\conf\Catali
```

7. Restart Tomcat.

WebMDS can then be configured and used as described in the rest of the documentation: WebMDS.

3. Backward compatibility summary

Protocol changes since GT version 4.0.x:

- None

API changes since GT version 4.0.x:

- None

Exception changes since GT version 4.0.x:

- None

Schema changes since GT version 4.0.x:

- None

4. Technology dependencies

WebMDS depends on the following GT components:

- Java WS Core

WebMDS depends on the following 3rd party software:

- Tomcat[4]

[4] http://jakarta.apache.org/tomcat/

5. WebMDS Security Considerations

By default, the WebMDS plugins distributed as part of the Toolkit do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information.

The `ResourcePropertyNodeSource` and `ResourcePropertyQueryNodeSource` plugins can be configured either to allow users to specify what resources they want to query or to only allow users to query resources pre-configured by the web administrator. The standard WebMDS deployment allows users to specify the resources they want to query; to disallow this (for example, to ensure that people don't use your site's bandwidth to view information about some other site's services), remove the files `$GLOBUS_LOCATION/lib/webmds/conf/openEndedRP` and `$GLOBUS_LOCATION/lib/webmds/conf/openEndedQuery`.

DRAFT

Chapter 2. Usage scenarios

There is no "client" programmatic interface to WebMDS; clients communicate using HTTP requests. The web form arguments recognized by WebMDS are documented in User's Guide.

1. Creating a new plugin

To create a new plugin to collect raw XML data, write a Java class that implements the `WebmdsXmlSource` or `WebmdsNodeSource` interface. These are documented in APIs. The `FileXmlSource` and `NodeXmlSource` classes distributed with WebMDS are examples of classes that implement `WebmdsXmlSource`; the `ResourcePropertyNodeSource` and `ResourcePropertyQueryNodeSource` classes distributed with WebMDS are examples of classes that implement the `WebmdsNodeSource` interface.

2. Changing format of output

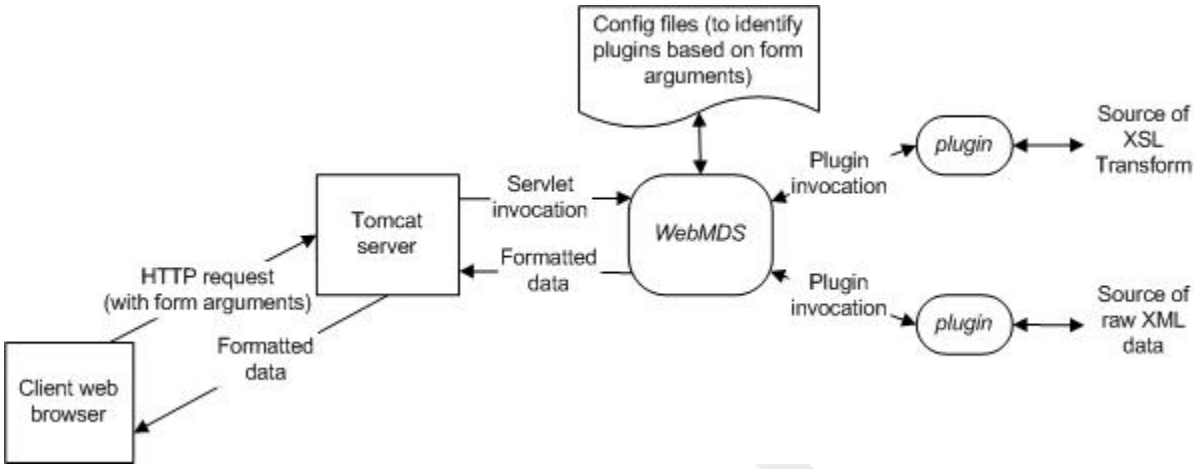
To change the appearance of the output of WebMDS, create a new XSLT transform; see the W3C XSLT Documentation[1] for more information.

DRAFT

[1] http://www.w3.org/TR/xslt

Chapter 3. Architecture and design overview

Figure 3.1. WebMDS Request Flow



In a typical WebMDS transaction, a user uses a web browser to send an HTTP request, including some web form arguments, to a web server / servlet container. The web server invokes the WebMDS servlet, which uses the form arguments to determine what plugins to use to retrieve the requested XML data and the XSLT transform to apply to it. The WebMDS servlet passes arguments to the plugins, which then retrieve the appropriate data and XSLT transform. The WebMDS servlet applies the XSLT transformation to the XML data and returns the result to the web server, which sends it back to the client's web browser.

DRAFT

Chapter 4. APIs

1. Programming Model Overview

There is no "client" API for accessing WebMDS; WebMDS is a servlet that is accessed via web forms.

WebMDS uses a *WebMDS plugin* (a Java class that implements the `WebmdsXmlSource` interface) to acquire XML documents (which can be used either as raw information sources or as XSL transformations). WebMDS comes with two WebMDS plugins: `FileXmlSource`, which reads XML from a file (and is primarily used to acquire XSL transformations), and `NodeXmlSource`. `NodeXmlSource` in turn calls a *node source plugin* (a Java class that implements the `WebmdsNodeSource` interface) to acquire an XML DOM document. acquires XML information using a *WebMDS XML source*, a Java class that implements the `WebmdsXmlSource` interface. To summarize:

- WebMDS is a servlet that uses plugins to acquire XML documents containing raw data and XSL transformations, and then applies the acquired XSL transformation on the acquired data.
    - The plugins used by WebMDS implement the `org.globus.mds.webmds.WebmdsXmlSource` interface.
    - WebMDS plugins include:
        - `org.globus.mds.webmds.xmlSources.file.FileXmlSource`, which reads XML from a file, and
        - `org.globus.mds.webmds.xmlSources.xmlDomNode.NodeXmlSource`, which uses its own plugin interface to acquire XML DOM documents.
            - The plugins used by `NodeXmlSource` implement the `org.globus.mds.webmds.xmlSources.xmlDomNode.WebmdsNodeSource` interface
            - Node source plugins include `org.globus.mds.webmds.xmlSources.resourceProperties.ResourcePropertyNodeSource` and `org.globus.mds.webmds.xmlSources.resourceProperties.ResourcePropertyQueryNodeSource`, which acquire resource property information.
- The raw XML data acquired by WebMDS is processed by XSL transformations; see the W3C XSLT Documentation[1] for more information on creating XSL transforms.

2. Component API

- Core WebMDS documentation[2] (includes the WebMDS servlet and the `WebmdsNodeSource` interface)
- FileXMLSource documentation[3]
- NodeXmlSource documentation[4] (including the `WebmdsNodeSource` interface)
- Resource property node source plugins[5].

[1] http://www.w3.org/TR/xslt
[2] http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_webmds/
[3] http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_webmds_file_source/
[4] http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_webmds_xml_dom_source/
[5] http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_webmds_resource_property_source/

Chapter 5. WebMDS Admin Commands

There is no end-user command-line tool for WebMDS.

1. Tool description

The command-line tool `webmds-create-context-file` is used to create Tomcat configuration files needed to deploy WebMDS.

2. Command syntax

```
webmds-create-context-file [-f] tomcat_context_file
```

The *tomcat_context_file* argument is the location of the Tomcat configuration file defining the WebMDS context; in a default Tomcat installation, the location of this file will be `$CATALINA_HOME/conf/Catalina/localhost`.

By default, `webmds-create-context-file` will not overwrite an existing context file; the `-f` option is used to force `webmds-create-context-file` to overwrite an existing file.

Note: `webmds-create-context-file` is found in `$GLOBUS_LOCATION/lib/webmds/bin`

3. Example

```
$GLOBUS_LOCATION/lib/webmds/bin/webmds-create-context-file -f \
     $CATALINA_HOME/conf/Catalina/localhost
```

4. Limitations

Changes to the Tomcat context do not take effect until Tomcat is restarted or reloaded.

DRAFT

Chapter 6. Graphical User Interface

1. Overview of the purpose and functionality of the GUI

The WebMDS GUI is a web-based interface for browsing formatted XML data, such as the results of resource property queries on a grid service.

2. Command and options

WebMDS can be accessed using any web browser. In a default WebMDS installation, the URL `http://host-name:port/webmds` corresponds to the top-level WebMDS web page. This page includes a link to a WebMDS invocation that provides summary information (with links to detailed information) about a locally-running MDS Index server. It also contains a link to a page of sample web forms demonstrating other uses of WebMDS.

3. Customizing the web forms used to access WebMDS

The WebMDS servlet is located at `http://your-tomcat-host:your-tomcat-port/webmds/webmds`. It takes the following arguments:

Table 6.1. Form arguments used by WebMDS

`info`	The name of the XML source that will be used to collect the raw XML data. XML sources are defined by files in `$GLOBUS_LOCATION/lib/webmds/conf`. This argument must be specified.
`xsl`	The name of the XML source that will provide the XSL transform. XML sources are defined by files in `$GLOBUS_LOCATION/lib/webmds/conf`. If this argument is not specified, the WebMDS servlet will display raw, untransformed XML.
`xmlSource.info_name.param.source_specific_options`	Any additional options recognized by the *info_name* XML source (*info_name* must be the value of the *info* argument for this request). Source-specific options are discussed in the next section.
`xmlSource.xsl_name.param.source_specific_options`	Any additional options recognized by the *xsl_name* XML source (*xsl_name* must be the value of the *xsl* argument for this request). Source-specific options are discussed in the next section.

4. Limitations

Error conditions (such as typographical errors in resource property names) are presented as stack traces, rather than user-friendly error messages.

Chapter 7. Configuring

1. Configuration overview

WebMDS can be configured to get information from any of various sources and to filter it through any XSL transform. WebMDS uses configuration files to specify the location of (and to name) sources of information and xsl and web form arguments to select among these configured information sources and xsl transforms.

By default, WebMDS comes configured to report information about an index server using transaction-level security on the default port (8443) on the local system. If you are running the Globus Toolkit in this default configuration, then you can use WebMDS to query your local *Index Service* without any configuration changes.

If you wish to monitor a different Index Service, you will need to edit the file `$GLOBUS_LOCATION/lib/webmds/conf/indexinfo` to change the URL in the line:

```
<value>https://127.0.0.1:8443/wsrf/services/DefaultIndexService</value>
```

to match the URL of your default index service. Changes to WebMDS configuration files take effect the next time that Tomcat is restarted.

For other configuration changes (e.g., monitoring different kinds of services), see the detailed configuration information below.

2. Syntax of the interface

Each configuration file in `$GLOBUS_LOCATION/lib/webmds/conf` defines a source of XML, which can be used in an HTML form to specify sources of information and XSL transforms. The distribution contains some standard configuration files in this directory, including:

Table 7.1. Pre-configured information sources

`indexinfo`	all resource properties from an index server running with transaction-level security on port 8443 on the local host
`indexinfo_nosec`	all resource properties from an index server running with no security on port 8080 on the local host
`openEndedQuery`	all resource properties from a user-specified grid service
`openEndedRP`	a user-specified resource property from a user-specified grid service
`servicegroupxsl`	an xsl transform that presents summary information about a service group
`sgedetail`	an XSL transform that presents detailed information about a service group entry

Each configuration file defines a `WebmdsConfig` object. A `WebmdsConfig` object consists of:

- A `description`: a textual description of the XML source being defined.
- A `className`: the name of the Java class that will be used to acquire the XML data.
- Zero or more `parameter` objects, each of which consists of the `name` of some parameter recognized by the Java class specified by `className`, and the string `value` of that parameter.

For example, this is `$GLOBUS_LOCATION/lib/webmds/conf/servicegroupxsl`, which defines the `servicegroupxsl` XML source:

```
<WebmdsConfig>
  <description>
     XSL file to show service group summary information
  </description>
  <className>org.globus.mds.webmds.xmlSources.file.FileXmlSource</className>
  <parameter>
    <name>file</name>
    <value>xslfiles/servicegrouptable.xsl</value>
  </parameter>
</WebmdsConfig>
```

This file tells WebMDS to use the `org.globus.mds.webmds.xmlSources.file.FileXmlSource` Java class (a class which reads XML from a local file) to collect XML data and to pass a `file` parameter (which that Java class interprets as the name of the file to open, relative to the WebMDS base directory).

Tomcat must be restarted (or one of the more advanced Tomcat administrative mechanisms must be used) for changes to these configuration files to take effect.

3. XML Sources included with WebMDS

3.1. FileXMLSource

The class `org.globus.mds.webmds.xmlSources.file.FileXmlSource` reads XML from a file, and recognizes a single parameter:

Table 7.2. Configuration parameters used with FileXMLSource

`file`	The name of the file to read. Relative path names are interpreted relative to the WebMDS base directory (`$GLOBUS_LOCATION/lib/webmds`).

3.2. NodeXMLSource

This XML source class uses a WebmdsNodeSource object to fetch an XML document and return it in a form that is usable by WebMDS. It recognizes the following options:

Table 7.3. Configuration parameters used with NodeXMLSource

`class`	The name of a class that implements the WebmdsNodeSource interface. An instance of this class will be used to get an XML document.
parameters	Additional parameters are passed to an instance of the class specified by the `class` argument.

3.3. Classes That Implement WebmdsNodeSource

The following classes implement the `NodeXMLSource` interfaces and can be used in conjunction with `NodeXMLSource`

3.4. ResourcePropertyQueryNodeSource

This class performs a resource property query to get all the resource properties for some web service. It recognizes the following configuration parameters:

Table 7.4. Configuration parameters used with ResourcePropertyQueryNodeSource

`endpoint`	The endpoint name to be used in a resource property query.
`endpointKeyName` and `endpointKeyValue`	An optional key/value pair to use as reference properties for the endpoint specified with the `endpoint` parameter.
`allowUserEndpoints`	If `true`, values for `xmlSource.sourceName.param.endpoint`, `xmlSource.sourceName.param.endpointKeyName`, and `xmlSource.sourceName.param.endpointKeyValue` specified in the request will override the configured endpoint value.
`endpointFile`	The name of a file from which the endpoint information (in XML) will be read. This configuration parameter can never be overridden by request arguments.

3.5. ResourcePropertyNodeSource

This class queries a web service for a single resource property. It recognizes the following parameters:

Table 7.5. Configuration parameters used with ResourcePropertyNodeSource

`endpoint`	The endpoint name to be used in a resource property query.
`endpointKeyName` and `endpointKeyValue`	An optional key/value pair to use as reference properties for the endpoint specified with the `endpoint` parameter.
`allowUserEndpoints`	If `true`, values for `xmlSource.sourceName.param.endpoint`, `xmlSource.sourceName.param.endpointKeyName`, and `xmlSource.sourceName.param.endpointKeyValue` specified in the request will override the configured endpoint value.
`endpointFile`	The name of a file from which the endpoint information (in XML) will be read. This configuration parameter can never be overridden by request arguments.
`rpNamespace`	The namespace part of the QName of the resource property to be queried for.
`rpName`	The local name part of the QName of the resource property to be queried for.
`allowUserResourceProperties`	If `true`, values of `xmlSource.sourceName.param.rpNamespace` and `xmlSource.sourceName.param.rpNames` specified in the request will override the configured resource property namespace and name.

Chapter 8. Debugging

For information on sys admin logs, see Chapter 5, Debugging.

Log information from WebMDS and any WebMDS plugins will be logged by the servlet container into which WebMDS has been deployed. In a vanilla Tomcat 5.0.28 distribution, this information will show up in the file `$CATALINA_HOME/logs/catalina.out`.

DRAFT

Chapter 9. Troubleshooting

For a list of common errors in GT, see Error Codes.

1. Error Messages

Error handling in WebMDS is currently done by throwing exceptions, which are displayed by Tomcat as stack traces.

Table 9.1. WS MDS Trigger Service Error Messages

Error Code	Definition

`java.net.ConnectException: Connection refused`	If you attempt to use WebMDS to collect information from a service that is not running, you will see a stack trace th
	
	`org.globus.mds.webmds.xmlSources.resourceProperties.ResourcePropertySourceExc`
	`java.net.ConnectException: Connection refused`

`faultString: org.globus.common.ChainedIOException: Authentication failed [Caused by: Failure unspecified at GSS-API level [Caused by: Unknown CA]]`	When WebMDS sends resource property queries to a secure WSRF service instance (such as an WS MDS Index Se ficate authority that issued the certificate used by the WSRF service instance. If the WebMDS server does not trust the queries will produce a stack trace that includes this message.
WebMDS connections to secure Index Servers (or other secure WSRF servers) just hang	If the JVM used by Tomcat is configured to use a blocking random-number source, WebMDS connections to secure can hang. This is the default configuration for many installations.

DRAFT

Chapter 10. Related Documentation

None available at this time.

DRAFT

Glossary

I

Index Service	An aggregator service in WS MDS that serves as a registry similar to UDDI, but much more flexible. Indexes collect information and publish that information as WSRF resource properties.

DRAFT

Index

DRAFT