

# **GT 4.2.1 Component Guide to Public Interfaces: WS MDS WebMDS**

---

## **GT 4.2.1 Component Guide to Public Interfaces: WS MDS WebMDS**

---

---

# Table of Contents

1. Graphical User Interface .....	1
1. Overview of the purpose and functionality of the GUI .....	1
2. Command and options .....	1
3. Customizing the web forms used to access WebMDS .....	1
4. Limitations .....	1
2. Configuring .....	2
1. Configuration overview .....	2
2. Syntax of the interface .....	2
3. XML Sources included with WebMDS .....	3
3. WebMDS Admin Commands .....	5
1. Tool description .....	5
2. Command syntax .....	5
3. Example .....	5
4. Limitations .....	5
4. APIs .....	6
1. Programming Model Overview .....	6
2. Component API .....	6
A. Errors .....	7
Glossary .....	8

---

## List of Tables

1.1. Form arguments used by WebMDS .....	1
2.1. Pre-configured information sources .....	2
2.2. Configuration parameters used with FileXMLSource .....	3
2.3. Configuration parameters used with NodeXMLSource .....	3
2.4. Configuration parameters used with ResourcePropertyQueryNodeSource .....	4
2.5. Configuration parameters used with ResourcePropertyNodeSource .....	4
A.1. WS MDS Trigger Service Error Messages .....	7

---

# Chapter 1. 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 1.1. Form arguments used by WebMDS**

<code>info</code>	The name of the XML source that will be used to collect the raw XML data. XML sources are defined by files in <code>\$GLOBUS_LOCATION/lib/webmds/conf</code> . This argument must be specified.
<code>xsl</code>	The name of the XML source that will provide the XSL transform. XML sources are defined by files in <code>\$GLOBUS_LOCATION/lib/webmds/conf</code> . If this argument is not specified, the WebMDS servlet will display raw, untransformed XML.
<code>xml-Source.info_name.param.source_specific_options</code>	Any additional options recognized by the <i>info_name</i> XML source ( <i>info_name</i> must be the value of the <i>info</i> argument for this request). Source-specific options are discussed in the next section.
<code>xml-Source.xsl_name.param.source_specific_options</code>	Any additional options recognized by the <i>xsl_name</i> XML source ( <i>xsl_name</i> must be the value of the <i>xsl</i> 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 2. 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/webm-ds/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/webm-ds/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 2.1. Pre-configured information sources**

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

Each configuration file defines a `Webm-dsConfig` object. A `Webm-dsConfig` 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/servicegroupxml`, which defines the `servicegroupxml` 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.xml</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 2.2. Configuration parameters used with FileXMLSource**

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

### 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 2.3. Configuration parameters used with NodeXMLSource**

<code>class</code>	The name of a class that implements the <code>WebmdsNodeSource</code> interface. An instance of this class will be used to get an XML document.
<code>parameters</code>	Additional parameters are passed to an instance of the class specified by the <code>class</code> 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 2.4. 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 <code>xmlSource.sourceName.param.endpoint</code> , <code>xmlSource.sourceName.param.endpointKeyName</code> , and <code>xmlSource.sourceName.param.endpointKeyValue</code> 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 2.5. 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 <code>xmlSource.sourceName.param.endpoint</code> , <code>xmlSource.sourceName.param.endpointKeyName</code> , and <code>xmlSource.sourceName.param.endpointKeyValue</code> 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 <code>xmlSource.sourceName.param.rpNamespace</code> and <code>xmlSource.sourceName.param.rpNames</code> specified in the request will override the configured resource property namespace and name.

---

# Chapter 3. 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.

---

# 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](#)<sup>1</sup> for more information on creating XSL transforms.

## 2. Component API

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

---

<sup>1</sup> <http://www.w3.org/TR/xslt>

<sup>2</sup> [http://www.globus.org/api/javadoc-4.0.0/globus\\_wsrf\\_mds\\_webmds/](http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_webmds/)

<sup>3</sup> [http://www.globus.org/api/javadoc-4.0.0/globus\\_wsrf\\_mds\\_webmds\\_file\\_source/](http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_webmds_file_source/)

<sup>4</sup> [http://www.globus.org/api/javadoc-4.0.0/globus\\_wsrf\\_mds\\_webmds\\_xml\\_dom\\_source/](http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_webmds_xml_dom_source/)

<sup>5</sup> [http://www.globus.org/api/javadoc-4.0.0/globus\\_wsrf\\_mds\\_webmds\\_resource\\_property\\_source/](http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_mds_webmds_resource_property_source/)

---

# Appendix A. Errors

**Table A.1. WS MDS Trigger Service Error Messages**

Error Code	Definition
<code>java.net.ConnectException: Connection refused</code>	If you attempt to use WebMDS to collect information from a service that is not running, you will see a stack trace that includes the following: <code>org.globus.mds.webmds.xmlSources.resourceProperties.ResourcePropertySourceException: java.net.ConnectException: Connection refused</code>
<code>faultString: org.globus.common.ChainedIOException: Authentication failed [Caused by: Failure unspecified at GSS-API level [Caused by: Unknown CA]]</code>	When WebMDS sends resource property queries to a secure WSRF service instance (such as an WS MDS Index Server) that does not have a certificate authority that issued the certificate used by the WSRF service instance. If the WebMDS server does not trust the certificate, 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 services can hang. This is the default configuration for many installations.

---

# 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.