Remote Database Authentication Server
(RDBAuth)
Administrator Guide

Revision: September 27, 2002

Table of Contents

Overview

The Remote Database Authentication Server (RDBAuth) controls and manages user access privileges to database resources for Morris Digital Works' applications and utilities based on user-defined attributes, business rules, and security policies. RDBAuth is the authorization server for applications built upon Morris Digital Works' Agent Framework. The RDBAuth server consists of multiple configuration files that allow the administrator to configure the server. Any modifications to these configuration files are permanent and should be handled with care. This manual will explain each configuration file and it's function.

How RDBAuth works

RDBAuth runs as a daemon and awaits requests from Morris Digital Works' clients. Each requesting client provides a service specific username and password, which RDBAuth will attempt to validate. If the username and password exist for the desired service, then RDBAuth relays to the client the connection string along with the database username and password for the access level of the requesting user. The client will then use the provided information to establish a connection directly to the database. This connection will be limited based upon the permissions setup for the user account provided.

Installing RDBAuth

Supported Platforms

- HP-UX 11
- Redhat Linux 6.2

System Requirements

Installation Requirements

- Java VM 1.3 or later
- JDBC Compliant Driver
- 30 MB free disk space
- 8 MB for initial install
- 20 MB for log files
- An IP Address and 2 available ports

User Requirements

- Microsoft Windows version: 98, 98 Second Edition, Millennium Edition, NT 4.0 with Service Pack 6 or later, 2000, or XP
- Macintosh OS X

Installation

The following table lists the supported platforms and their corresponding archives.

Platform	Installation Archive
HP-UX 11	rdbauth_2.0_hpux11_bin.tar.gz
Redhat Linux 6.2	rdbauth_2.0_linux-x86_bin.tar.gz

Supported platforms and their corresponding installation archive.

Before you begin installation, you will need to insert the CD entitled "mdClassifieds Development Kit" in your CD drive. Once you've done this, navigate to the readme.txt file located at the root level of the CD drive. The readme.txt file will contain all the information you will need concerning the location of the product installer. Once you find the correct installer location, follow these example instructions on how to extract the archive for the HP-UX 11 platform and relocate the files.

```
$ gunzip rdbauth_2.0_hpux11_bin.tar.gz
$ tar -xvf rdbauth_2.0_hpux11_bin.tar
$ mv rdbauth_2.0_hpux11_bin /opt/rdbauth/
```

Configuration

The RDBAuth Server installation comes standard with 4 configuration files. This section will describe how to quickly get the server up and running.

The following configuration files are located here: /opt/rdbauth/conf/. In order to make permanent changes to the configuration files, you'll need to cd into the above directory and edit the file. When done, save the changes.

Configuring rdbauth.xml

The rdbauth.xml configuration file contains settings for the RDBAuth Server Component. You will notice that the configuration file is in XML format, each tag represents a different setting or set of settings. The first setting you will want to edit is the `<address>` tag, it should look like this by default:

```
<address>ip address</address>
```

You will want to replace “ip address” with either the domain name or ip address that you wish this server to listen on. The term "listen on" means that the RDBAuth server has a specific ip address and port open, and it is waiting for a request from a client. For a client to connect to the RDBAuth server, the server is required to be listening on an ip address and port that the client has access to. If the client does not have access to the specified ip address and port, then the client's request cannot be processed.

The next series of settings resides within the `<jdbc_pool>` tag. These setting dictate the behavior of the JDBC Connection Pooling Mechanism within the server; connection pooling is used to recycle the connections to the database in an efficient manner. There are only two settings that require editing to get the connection pool to work properly. The first setting is the `<driver>` tag, which specifies the Java Class name of the driver to be used. By default the Oracle JDBC Driver is used. If you are using a DBMS Vendor other than Oracle, then it is recommended you refer to the vendor’s documentation for the classname of their driver.

The next <jdbc_pool> setting is the <url> tag. This specifies the connection string for connecting to your database. The tag is already preformatted for use with the Oracle JDBC Driver. If you are using Oracle as your DBMS, then you will only have to replace the following pieces of the URL:

ip address – replace with the domain name or IP address of your database server.
port – replace with the port number your database server is listening on.
service – replace with the service name of your database.

The remaining settings are set to their default values. These settings are all defined within *Administrator Maintenance* with their descriptions, default values and allowed values.

Configuring http.xml

The http.xml configuration file contains settings for the HttpServer Component. This is the configuration file that determines administrative settings. In this file, you will determine the security settings, who can have administrative privileges, and usernames and passwords. The http server is configurable via four sets of settings: <listeners>, <hosts>, <users>, and <security>.

The <listeners> tag contains a list of addresses and ports that the HttpServer will listen on for administration requests. The <listener> tag has two required settings, address and port. For example, a <listener> tag looks like this:

```
<listener address="ip address" port="port" />
```

You can listen on the same IP Address you defined in rdbauth.xml so long as you use a different port number. So if you wanted to listen on IP address127.0.0.1 and port 1234, you would edit the tag as follows:

```
<listener address="127.0.0.1" port="1234" />
```

The next set of settings is the <hosts> tag. This tag specifies a list of IP Addresses allowed to access the server. Each address is contained within a <host> tag. You would add a <host> tag for each IP Address (or range of addresses) that you want to have administration privileges. So for instance, if you wanted the computer that the server is residing on to have access to the server, you would add the following <host> tag:

```
<host address="127.0.0.1"/>
```

The <users> tag is another built-in security layer. This specifies a list of users who are allowed to access the server. Each tag defines a username and a password. An example user would be:

```
<user username="admin" password="hello"/>
```

A user login is only required when the resource they are attempting to access is secure. A secure resource is defined within the <security> tag. This setting may contain multiple <secure> tags. By default, the following tag is defined:

```
<secure path="*"/>
```

The path="*" setting specifies that all paths on the server are secured. The path setting should be the path to a resource which is available and which you want password protected. The RDBAuth server by default only has one available path, which is /jmx, this resource is for the management of the server. Because path="*", only valid users can access /jmx. Also, if you set path="/jmx", this will also only allow valid users.

The remaining configuration files and setting are explained within *Administrator Maintenance*. Please refer to *Administrator Maintenance* for detailed descriptions, default settings and allowed settings.

Startup and Shutdown

Before you can start the server, you first need to modify the start and stop scripts. These scripts can be found at /opt/rdbauth/bin/. In order to make permanent changes to the start and stop scripts, you'll need to cd into the above directory and edit the script. When done, save the changes.

Startup Script

The startup script is simply named "start". The first line of the script specifies the location of the install path of the Java VM 1.3. By default, the value looks like:

```
JAVA_HOME=/usr/java/jdk1.3.1_02
```

If your install path is different, please change the path for `JAVA_HOME`. The next line should specify the install path of the RDBAuth Server. By default, the value looks like:

```
RDBAUTH_HOME=/opt/rdbauth
```

If your install path is different, please change the path for `RDBAUTH_HOME`.

The usage of the start script is:

```
bin/start [username] [password]

[username] - the username for connecting to the database
[password] - the password for connecting to the database
```

For example, if you wanted to have the server use the username scott and password tiger as its login then you would run:

```
$ bin/start scott tiger
```

Shutdown Script

The shutdown script is simply named "stop". The first line of the script specifies the install path of the RDBAuth Server. By default, the value looks like:

```
RDBAUTH_HOME=/opt/rdbauth
```

If your install path is different, please change the path for `RDBAUTH_HOME`.

To run the script, simply call:

```
$ bin/stop
```

Administrator Maintenance

The following configuration files are located at /opt/rdbauth/conf/. In order to make permanent changes to the configuration files, you'll need to cd into the above directory and edit the file. When done, save the changes.

Configuration files: http.xml

Detailed definitions and allowed settings for the HTTP Server.

<http>
This tag is the root tag for the HTTP Server configuration. The <http> tag contains the following tag sets:

- <listener>
- <hosts>
- <security>
- <users>

<listeners>
This tag contains a list of listeners. A listener defines the IP addresses and ports that it will listen for requests on. You would add a listener for each address and port combination that you want the server to listen on.

```
<listeners>
      <listener address="[IP Address]" port="[port]" />
</listeners>
```

<hosts>
This tag contains a set of allowed hosts. The host defines the IP Address of an allowed computer. If a host is not defined in the list, then it is not allowed to access the HTTP Server. You would add a host for each IP Address you wish to have access to the server. You may also enter a range of IP Addresses. So if you wanted to add hosts in the IP range 127.0.0.1 to 127.0.0.255, you could edit the tag as following:

```
<hosts>
      <host address="127.0.0.1-127.0.0.255"/>
</hosts>
```

When entering a range of IP Addresses, you must follow these rules:

- Valid range of IP Addresses
- Dash is required between a valid starting IP Address and a valid ending ip address
- First three octets must be the same (in the example above, 127.0.0)
- Fourth octet of the first IP Address must be less than the fourth octet of the second address (in the above example, 1 < 255)

<security>
This tag contains a set of secure paths. The <secure> tag defines a path that is to be password protected. Only users defined in the <users> set will be allowed to access any of the secure paths defined here.

```
<security>
      <secure path="[URI path]" />
</security>
```

For example, if the server's URL is `http://127.0.0.1/private`, and you wanted the URI of `/private` to be password protected, then you would add the following secure tag:

```
<secure path="/private" />
```

If you wanted all paths secure, then you would add the following tag:

```
<secure path="*" />
```

<users>
This tag contains a set of allowed users. <users> defines the username and password of each user allowed to access the HTTP server at any of the secure paths listed in the <security> set. Any user not defined in the <users> list will not gain access to the secure paths under <security>.

```
<users>
      <user username="[username]" password="[password]" />
</users>
```

Configuration files: rdbauth.xml

Detailed definitions, default values, and allowed values for the RDBAuth Server. The default setting is acquired only if no value is specified. For example, say the <address> tag is set as follows:
<address></address>
The <address> tag is empty, therefore a value is not specified, and the value of the <address> tag defaults to 127.0.0.1.

<config>
This tag is the root tag for the RDBAuth Server configuration. The <config> tag contains the following tag sets:

- <address>
- <port>
- <invalid_max>
- <invalid_exp>
- <stored_proc>

<address>
This tag contains the IP Address for the server.

```
<address>127.0.0.1</address>
```

Valid Values: IP Address, Domain Name
Default Value: 127.0.0.1

<port>
This tag specifies the port to listen for requests on the address specified in <address>.

```
<port>16363</port>
```

Valid Values: Valid Ports: a number greater than 1024
Default Value: None

<invalid_max>
The maximum number of failed attempts before the user is blocked.

```
<invalid_max>5</invalid_max>
```

Valid Values: 0-255
Default Value: 5

<invalid_exp>
The time (in seconds) it will take for a blocked user to be allowed access.

```
<invalid_exp>5</invalid exp>
```

Valid Values: any integer value greater than or equal to 0 (zero)
Default Value: 5

<stored_proc>
The stored procedure in the database used to authenticate users.

```
<stored_proc>rdbauth.get_service_attribute</stored_proc>
```

Valid Values: The name of the stored procedure. The stored procedure must have the following parameters:

1. Service Name (string)
2. Service Attribute Name (string)
3. Username (string)
4. Password (string)

Default Value: rdbauth.get_service_attribute

<jdbc_pool>
Configuration settings for the JDBC Connection Pool used with in the RDBAuth Server. This tag is contains the following tag set:

- <description>
- <max_size>
- <min_size>
- <clean_interval>
- <clean_scheme>
- <cached>
- <scheme>
- <log_stdout>
- <driver>
- <url>

<description>
Provides general description of the pool.

```
<description>My JDBC Pool</description>
```

Valid Values: ANY
Default Value: NONE
Required: NO

<max_size>
The maximum number of database connections to have opened at a single time.

```
<max_size>5</max_size>
```

Valid Values: any integer value greater than 0 (zero)
Default Value: 5

<min_size>
The minimum number of database connections to keep open.

```
<min_size>1</min_size>
```

Valid Values: any integer value greater than or equal to 0 (zero)
Default Value: 1

<clean_interval>
The time interval (in seconds) for the pool cleaner to validate connections within the pool and to close invalid connections and connections, which are no longer used.

```
<clean_interval>180</clean_interval>
```

Valid Values: any integer value greater than 0 (zero)
Default Value: 180

<clean_scheme>
The clean scheme defines the method for the pool cleaner to close connections no longer used.

```
<clean_scheme>SINGLE_CLEAN</clean_scheme>
```

Valid Values:
- `SINGLE_CLEAN` - Cleans out one connection per `<clean_interval>`
- `BATCH_CLEAN` - Cleans all unused connections at each clean_interval

Default Value: `SINGLE_CLEAN`

<cached>
Determines caching scheme for the connections used within the pool. If caching is turned off then each authentication request will be made using a new database connection, this however will lead to slow response times because of the time required in opening connections.

```
<cached>false</cached>
```

Valid Values:
- `TRUE` - Cache database connections for reuse
- `FALSE` - Do not reuse database connections.

Default Value: `FALSE`

<scheme>
Determines the behavior of the pool when a database connection is not immediately available upon request.

```
<scheme>FIXED_WAIT</scheme>
```

Valid Values:
- `FIXED_WAIT` - Wait until a connection is available
- `FIXED_NO_WAIT` – Do not wait until a connection is available.

Default Value: `FIXED_WAIT`

<log_stdout>
FOR DEBUGGING PURPOSES ONLY. If set to true, flushes all JDBC Connection Pool logging statements to STD_OUT.

```
<log_stdout>false</log_stdout>
```

Valid Values:
- `TRUE` - flush logging messages to STD_OUT
- `FALSE` - does not flush logging messages to STD_OUT.

Default Value: `FALSE`

<driver>
Defines the class name of the JDBC driver used to connect to the database.

```
<driver>oracle.jdbc.driver.OracleDriver</driver>
```

Valid Values: Qualified Class Name of the JDBC Driver
Default Value: NONE
Required: YES

<url>
Defines the JDBC Connection URL for establishing a connection to the database.

```
<url>jdbc:oracle:thin:@address:port:service</url>
```

Valid Values: Qualified JDBC URL for connecting to the database
Default Value: NONE
Required: YES

Error logs

There are two error files located at /opt/rdbauth/logs:

agent.log
This log file contains information regarding the status of the server (running or stopped). This file also details which components are loaded in the server.

rdbauth.log
This log file contains information about users that have attempted to authenticate, their status, the time they tried to authenticate, and their IP address.

Program Recovery

If you experience trouble with the rdbauth server, stop the server and then restart it. If you continue to have trouble, please contact customer support.

Support Information

Morris Digital Works (MDW), a division of Morris Communications Co., provides tools, technologies, consulting and Web development services to Morris newspapers and external clients. MDW award winning technologies include; world-class hosting facilities, robust content management software, high performance application tools, site enhancement tools and comprehensive classified and display classified technology. Founded in 1995, Morris Digital Works has over 100 employees with offices in Topeka, KS, Joplin, MO, New York, NY and headquartered in Augusta, GA. An additional 250 MCC employees also participate in our Internet business and report directly to newspapers, magazines, book publishing and other internal organizations.

If you are experiencing problems with any of Morris Digital Works products, please contact customer support at (706) 828-2955 ext. 2.