



# Using logcontroller and useful log categories

Revision 20220519

## NOTE

This document is confidential and proprietary of **Denodo Technologies**.  
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2022  
Denodo Technologies Proprietary and Confidential

## CONTENTS

<b>1 GOAL.....</b>	<b>5</b>
<b>2 CONTENT.....</b>	<b>6</b>
<b>3 LOG CATEGORIES.....</b>	<b>7</b>
<b>3.1 NORTHBOUND CONNECTIONS.....</b>	<b>7</b>
<b>3.1.3 ENABLING LOGS FOR ODBC CONNECTIONS.....</b>	<b>7</b>
<b>3.1.4 ENABLING OAUTH LOGS FOR REST/SOAP WS PUBLISHED FROM VIRTUAL DATAPORT.....</b>	<b>8</b>
<b>3.2 SOUTHBOUND CONNECTIONS.....</b>	<b>8</b>
<b>3.2.4 TO GET THE DATA SOURCE ACCESS INFORMATION.....</b>	<b>9</b>
<b>3.3 AUTHENTICATION AND SECURITY.....</b>	<b>9</b>
<b>3.3.3 KERBEROS TROUBLESHOOTING.....</b>	<b>10</b>
<b>3.4 CACHE.....</b>	<b>10</b>
<b>3.4.1 ENABLE CACHE LOGS.....</b>	<b>10</b>
<b>3.4.2 BULK DATA LOAD.....</b>	<b>10</b>
<b>3.4.3 TABLE INSERTION.....</b>	<b>10</b>
<b>3.5 PERFORMANCE.....</b>	<b>11</b>
<b>3.5.1 EXECUTION TRACE.....</b>	<b>11</b>
<b>3.5.2 COST-BASED OPTIMIZER.....</b>	<b>11</b>
<b>3.6 VIRTUAL LAYER AND OTHER SCENARIOS.....</b>	<b>12</b>

**3.6.3 STORED PROCEDURES.....13**

**3.6.5 SOAP REQUESTS / RESPONSES.....13**

**4 REFERENCES.....14**

## 1 GOAL

---

This document explains how to use the predefined [LOGCONTROLLER](#) stored procedure and some useful log categories.

## 2 CONTENT

---

Denodo Virtual DataPort uses the logging library Apache Log4j to log the activity of the Virtual DataPort server. Log4j allows defining several categories on which the log level will be specified independently.

By default, the Virtual DataPort Server logs are stored in <DENODO\_HOME>/logs/vdp and the Administration Tool logs in <DENODO\_HOME>/logs/vdp-admin. The configuration of Log4j of the Virtual DataPort server is controlled by the file log4j2.xml (<DENODO\_HOME>/conf/vdp) and for the Administration Tool by the file log4j.xml (<DENODO\_HOME>/conf/vdp-admin). Refer to article [Log column details](#) to understand the content of the logs.

You can change the log level of a category in the VDP server without restarting it by calling the stored procedure "LogController". This stored procedure changes the log level for a certain log category of the logging engine used by Virtual DataPort at run time. The changes performed by this procedure are lost when the Server is stopped. Administration privileges are required to call this stored procedure.

Syntax: LogController (<logCategory>, <logLevel>)

- logCategory - the category of the log. The package of the Java classes to log information from. You can see some of the lists of log categories in log4j2.xml file.
- logLevel - type of message to be logged for the category: 'TRACE', 'DEBUG', 'INFO', 'WARN', 'ERROR', 'FATAL'.

## 3 LOG CATEGORIES

---

The log categories have a hierarchical structure and when a log level is set for a category, all its subcategories will also be set to the same log level.

Though all the classes come under the “com.denodo” category, it is not advisable to set the log level at this category as this would print all the log information of its subcategories, which in addition to occupying more space, could also delay the debugging process.

This document contains some useful scenarios categorized as Northbound Connections, Southbound Connections, Authentication, Cache and Performance that can be used during a troubleshooting process. **Note that**, after making changes to the log levels, it is recommended and important to revert back to the default level (i.e) ERROR after the troubleshooting is complete. In case you forgot the log levels set for different categories you can use the Predefined Stored Procedure [GET\\_ACTIVE\\_LOGGERS](#) that returns information about the log categories that have been set in the Virtual DataPort server.

### 3.1 NORTHBOUND CONNECTIONS

#### 3.1.1 Enable vdp-requests log

The vdp-requests.log will log information about all the query requests received by the Virtual DataPort. To set the log category for logging the requests, execute the below from the VQL shell

```
CALL LOGCONTROLLER('com.denodo.vdp.requests', 'INFO')
```

#### 3.1.2 Enable vdp-queries.log

The vdp-queries.log contains the VQL requests with more detailed information such as which user has executed what query, the total duration of the query, etc., and just like, vdp-requests.log, it has to be manually enabled by executing the below from the VQL shell.

```
CALL LOGCONTROLLER('com.denodo.vdb.management.mbeans.request.RequestsManagementInfo', 'INFO')
```

#### 3.1.3 Enabling logs for ODBC connections

When querying the Virtual DataPort from an ODBC client, the below combination of log categories can be set to debug any issues/ errors.

```
CALL LOGCONTROLLER('com.denodo.vdp.requests', 'INFO');  
CALL LOGCONTROLLER('com.denodo.vdb.vdbinterface.server.odbc', 'TRACE')
```

### 3.1.4 Enabling OAuth logs for REST/SOAP WS published from Virtual DataPort

The following log category can be set to diagnose problems when accessing the web services deployed in Denodo using OAuth2.0 authentication.

```
CALL LOGCONTROLLER('com.denodo.vdb.security.OauthAuthenticator', 'DEBUG')
```

## 3.2 SOUTHBOUND CONNECTIONS

### 3.2.1 To see the SQL sentence Denodo sends to a JDBC data source

The below log category can be set to log the prepared SQL sentence as seen in the execution trace of a view/ query execution.

```
CALL  
LOGCONTROLLER('com.denodo.vdb.engine.wrapper.raw.jdbc.adapter.DriverAdapter',  
, 'DEBUG')
```

### 3.2.2 Enabling error tuples log in Delimited Files Sources

The below log category can be set to log more information about query execution against a DF data source.

```
CALL  
LOGCONTROLLER('com.denodo.vdb.engine.wrapper.raw.df.DataExtractor.contentMat  
cher', 'INFO')
```

### 3.2.3 HTTP route

This log category can be set to debug HTTP data sources in order to see more details about the HTTP request such as headers etc.,

```
CALL LOGCONTROLLER('httpclient.wire','TRACE')
```

### 3.2.4 To get the Data Source access information

This log category can be used to get the data source access information. Replace [datasourcetype] with any of the below:  
arn, df, essbase, gs, html (itpilot), jdbc, json, ldap, custom, odbc, olap, salesforce, sapbwapi, saperp, ws or xml

```
CALL LOGCONTROLLER([datasourcetype], 'DEBUG') ('com.denodo.vdb.engine.wrapper.raw')
```

## 3.3 AUTHENTICATION AND SECURITY

### 3.3.1 Debugging LDAP Configuration

Refer to [Debugging the LDAP configuration](#) section from article [How to configure a VDP database with LDAP authentication and LDAP authentication best practices](#) for debugging steps

### 3.3.2 SSL related troubleshooting

First step is to add the property '-Djavax.net.debug=all' to the JVM configuration. To do that,

- Stop the Virtual Dataport Server
- Append the following property to the JVM configuration of Virtual DataPort Server using the Denodo Platform Control Center by going to Configure > JVM options > Virtual DataPort server or by editing the file <DENODO\_HOME>/conf/vdp/VDBConfiguration.properties at the property java.env.DENODO\_OPTS\_START

```
-Djavax.net.debug=all
```

- Execute the <Denodo\_Home>/bin/regenerateFiles.sh (.bat for Windows)
- Restart the Virtual DataPort Server.
- Then, open the VQLShell and execute the following log category

```
CALL LOGCONTROLLER('com.denodo.util.logging.JavaConsoleLogging', 'DEBUG')
```



### 3.3.3 Kerberos troubleshooting

Kerberos debug modes for Northbound, Southbound and most common issues are addressed in the article [Kerberos configuration and troubleshooting](#).

### 3.3.4 Privilege auditing

To debug the permissions assigned to a view for each user and role the following log category can be enabled:

```
CALL LOGCONTROLLER('com.denodo.vdb.security.queryPrivileges', 'DEBUG')
```

## 3.4 CACHE

### 3.4.1 Enable Cache logs

Use the following log category to enable cache logs

```
CALL LOGCONTROLLER('com.denodo.vdb.cache', 'DEBUG')
```

### 3.4.2 Bulk Data Load

Enable the following log category if Virtual Dataport runs into error during Bulk Data load. Execute the cache load query with necessary CONTEXT clauses after setting this log category.

```
CALL LOGCONTROLLER('com.denodo.vdb.util.tablemanagement.sql.insertion',  
'TRACE')
```

### 3.4.3 Table insertion

This combination of log categories can be set to debug the cache table insertion errors.

```
CALL LOGCONTROLLER('com.denodo.vdb.util.tablemanagement.TableManager',  
'DEBUG')  
CALL LOGCONTROLLER('com.denodo.vdb.util.tablemanagement.sql.insertion',  
'DEBUG')
```

### 3.4.4 Cache maintenance task

To debug any issues with the cache maintenance task, use the below log category

```
CALL LOGCONTROLLER('com.denodo.vdb.cache.CacheMaintainerTask', 'DEBUG')
```

### 3.5 PERFORMANCE

#### 3.5.1 Execution trace

To see the execution trace of the queries executed on the VDP server the following log category can be enabled. For more information see [Execution Trace of a Statement](#).

```
CALL LOGCONTROLLER('com.denodo.vdp.traces', 'DEBUG')
```

#### 3.5.2 Cost-based optimizer

For debugging the statistics generation process while gathering statistics for JDBC base view, the below log category can be set.

```
CALL LOGCONTROLLER('com.denodo.util.jdbc.sql', 'TRACE')
```

#### 3.5.3 Insufficient resources

This log category can be used to debug the scenarios where it says insufficient resources to execute an operation. For example, *java.lang.NoClassDefFoundError: Could not initialize class*

```
CALL LOGCONTROLLER('com.denodo.vdb.interpreter.execution.Action', 'WARN')
```

#### 3.5.4 Memory usage of a query execution

This log category can be set to log information about the memory usage of a query execution that can be useful in assessing the performance of a query. For example, in case, you are executing a query with a memory limit = 2000MB, you can retrieve information like,

```
Search method plan ready and selected to run: sm-0  
checkMemoryUsage - Max memory: 2.0 GB  
checkMemoryUsage - Query size: 0 B  
checkMemoryUsage - SwapGroups: 5
```

```
checkMemoryUsage - SwapSizes: [671088640, 637534208, 385875968, 201326592, 134217728]
checkMemoryUsage - Max swap size: 640.0 MB
checkMemoryUsage - Total queues: 945.0 MB
checkMemoryUsage - Query max size: 1.5 GB
checkMemoryUsage - Query memory usage below threshold
```

```
CALL LOGCONTROLLER('com.denodo.vdb.catalog.view.View', 'DEBUG')
```

### 3.5.5 Cost estimation

This combination of log categories can be set when cost optimization is applied during a view execution and the factual details about the optimizations applied.

```
CALL LOGCONTROLLER('com.denodo.vdb.engine.costoptimizer.LeafCostCalculator', 'DEBUG')
CALL LOGCONTROLLER('com.denodo.vdb.engine.costoptimizer.CostOptimizerQueryVisitor', 'DEBUG')
```

### 3.5.6 Transformations candidates costs

To see the final transformations candidates costs only applied to a query, use the log category

```
CALL LOGCONTROLLER('com.denodo.vdb.catalog.view.visitor.TreeViewTransformer', 'DEBUG')
```

## 3.6 VIRTUAL LAYER AND OTHER SCENARIOS

### 3.6.1 VCS

To debug any issues related to the Version control system, use the below log category

```
CALL LOGCONTROLLER('com.denodo.vdb.interpreter.execution.vcs', 'DEBUG')
```

### 3.6.2 SWAP

To get information about swapping use the below log category

```
CALL LOGCONTROLLER('com.denodo.vdb.engine.swap', 'DEBUG')
```

### 3.6.3 Stored Procedures

To enable stored procedure logs use the below log category

```
CALL LOGCONTROLLER('com.denodo.vdb.engine.storedprocedure', 'DEBUG' )
```

### 3.6.4 Compiling classes

When using the WSDL2Java Axis library, the WSDL file is translated into Java classes, which encapsulates the SOAP operations and messages. For obtaining the errors while compiling these classes, you can enable the following logging category,

```
CALL LOGCONTROLLER('com.denodo.util.compile', 'DEBUG')
```

### 3.6.5 SOAP Requests / Responses

**3.6.6 Denodo obtains tuples from a web service response (i.e., from the objects obtained from Apache Axis library/ from the underlying SOAP response message). For obtaining information about SOAP requests/responses, you can enable the following logging categories,**

```
CALL LOGCONTROLLER('org.apache.axis.client', 'DEBUG')  
CALL LOGCONTROLLER('org.apache.axis.transport.http', 'DEBUG')
```

### 3.6.7 Derby locks

Derby attempts to prevent two JVMs from accessing a database at one time by use of a db.lck file. In scenarios, where you face issues with the booting up of this database, you can debug the issue by setting the below log category

```
CALL LOGCONTROLLER('com.denodo.vdp.derby', 'DEBUG')
```

## 4 REFERENCES

---

Virtual DataPort Administration Guide: [Configuring the Logging Engine](#)  
Advanced VQL Guide: [LOGCONTROLLER](#)