Nebulae MultiUser Server® Help

For up-to-date information please visit our web site:
xtras.tabuleiro.com

NEBULAE MUTIUSER SERVER HELP: GETTING STARTED

Tabuleiro Nebulae MultiUser Server is a server application that runs on Linux, Solaris, MacOSX, Windows and almost any Java-enabled operational system. Nebulae is 100% compatible with the Shockwave MultiUser protocol published by Macromedia, so movies authored for the Shockwave MultiUser Server version 2 and 3 can connect seamlessly to a Nebulae server, without modifications.

Java is emerging as the strongest platform solution for server applications that need to service hundreds or thousands of users at the same time. Nebulae is 100% pure Java 2 code and works with JDK 1.5 or later. Nebulae provides an alternative to Shockwave developers that need to host multiuser movies in Unix systems, while preserving the investment made in learning the Shockwave MultiUser API.

All Shockwave MultiUser Server standard messages and commands can be used transparently with a Nebulae server, including DBObject functions. Nebulae includes a database engine so no database setup is required on the hosting machine, and database files are created and initialized automatically by the server. Nebulae also implement commands not available in the Shockwave MultiUser Server, like the ability to add banned user entries and ip addresses to the database. restart or shutdown the server remotely, send email and interface with standard SQL database engines.

The most important feature not supported by Nebulae in comparison with version 3 of the Shockwave MultiUser Server is server side Lingo scripting, since a Lingo interpreter is not available for other platforms. The server side scripting language used in Nebulae is Java: this is usually a more powerful solution for enterprise-level servers. Please click here for more information about server side scripting in Nebulae.

UDP protocol support is available when Nebulae is used with Director 8.5 or later and Shockwave applications.

We recommend checking the SYSTEM REQUIREMENTS section at the Nebulae page on xtras.tabuleiro.com for updated information on the latest Java virtual machines tested with Nebulae and performance considerations.

NEBULAE MUTIUSER SERVER HELP: INSTALLING THE SERVER

Tabuleiro Nebulae MultiUser Server is available as a packaged Java file, named "Nebulae.jar". Nebulae download packages also include a version of the free HSQL Database engine, used for the internal database functions. Nebulae servers can operate with or without database functions, but we recommend enabling database support in order to access administrative functions and DBObject commands. A third file named "Nebulae.cfg" is used to configure the server. There is also a template file named "Movie.cfg" that can be used to configure movie-specific commands.

Nebulae has been installed and tested in Solaris, FreeBSD, MacOSX, Windows and in different distributions of Linux (RedHat, Slackware and Debian). The only real requirement is a Java virtual machine compatible with Java 2 (JDK 1.5 or newer.)

Note: users upgrading from Nebulae 1.1 should consult the technote at xtras.tabuleiro.com/support/technotes/nebulae/ns007.htm for additional information before installing version 1.2.7. Users upgrading to version 2 should consult the technote at xtras.tabuleiro.com/support/technotes/nebulae/ns008.htm.

Installing on Linux/Solaris/Unix

Installing on MacOSX

Installing on Windows

After installation is complete you can customize the server by editing the "Nebulae.cfg" file. This file provides most of the configuration options also available for the Shockwave MultiUser Server 3, so the settings should be familiar to most multiuser developers:

Nebulae.cfg file configuration

It is also possible to configure movie-specific properties. A template file named "Movie.cfg" is supplied. In order to configure a specific movie you can should copy and rename this file to match the name of the target movie, for example to

"Chat.cfg" to configure a movie named "Chat". All configuration files should be kept in the same directory where the server application is started.

Movie.cfg file configuration

NEBULAE MUTIUSER SERVER HELP: INSTALLING THE SERVER

INSTALLING ON LINUX/UNIX/SOLARIS

Nebulae has been installed and tested in different distributions of Linux (RedHat, Slackware and Debian), Solaris and other *nix systems. We recommend using the the official Java VM from Sun if possible.

The Nebulae server is distributed in a .tar file available at the DOWNLOAD section on xtras.tabuleiro.com. We suggest that the files are decompressed in the /usr/local/nebulae directory, but you can keep them anywhere on your system. You can create a new user to run the server or run it as a root or any unprivileged account: Java applications rely on the Java VM to provide security to the whole system. The only restriction is that the user running the server must have access rights to the directory where the server files are installed, since it will be necessary to create database files and a server log. To create the directory (this assumes you are logged as root) and decompress your files open a shell account and issue the following commands:

```
cd /usr/local

mkdir nebulae
```

You can copy the Nebulae_2.tar file downloaded from Tabuleiro's site to this directory, and use the following command to decompress it:

```
cd nebulae

tar -xvf Nebulae_2.tar
```

This will create files Nebulae.jar, hsqldb.jar, Nebulae.cfg, ScriptMap.cfg and Movie.cfg. The original .tar file can be deleted at this time. You can also decompress the tar file with another utility like Winzip on your PC and upload the decompressed archives directly to your Linux server.

Before the server runs for the first time it is recommended that you create at least one administrative user in the server database, to allow access to server administration commands in the future. This can be accomplished by editing the

"CreateUser" directive in the Nebulae.cfg file:

```
#CreateUser = admin,pass,100
```

Uncomment the line by removing the pound sign at the beginning, and customize the values for username and password. This command will create a user names "admin", with password "pass" and userlevel 100 the first time the server is started. Please note that while the server can operate without a server administrator account you will not be able to shutdown it remotely until an administrative user is added. You can use pico or other text editor to edit the configuration file.

We recommend running the official Java VM from Sun. You can issue the following command to verify that the installation is functional:

```
java -version
```

Version 1.6 of Sun's VM will return the following output:

```
java -version

java version "1.6.0"

Java(TM) SE Runtime Environment (build 1.6.0-b105)

Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode, sharing)
```

To locate the path to java on your system, use:

```
which java
```

You can then start the Nebulae server and put it to run in the background with the following command:

```
/usr/java/jdk/bin/java -cp Nebulae.jar:hsqldb.jar net.tabuleiro.nebulae.Nebulae &
```

Please remember to run the command from the directory where Nebulae is installed. You can also use a shell script to start the server. This version adds the Nebulae directory to the classpath, so you can put server side script classes in the same directory of Nebulae (/usr/local/nebulae) in order to have them loaded by your movies. If you need to use a JDBC driver (for PostgreSQL connection for example) you should also remember to list its location on the classpath:

```
#!/bin/csh -f

cd /usr/local/nebulae

unlimit

/usr/java/jdk/bin/java -Dfile.encoding=ISO-8859-1 -cp
Nebulae.jar:hsqldb.jar:/usr/local/nebulae:postgresql.jar
net.tabuleiro.nebulae.Nebulae &
```

The "unlimit" command is useful on some distributions, where there is a limit on the number of file descriptors available per user. If you need to support more than 1024 simultaneous connections it may be necessary to raise the number of file descriptors available to your user session: please consult your Linux distributor for more assistance on the commands appropriated for your specific distribution.

For your convenience Nebulae can be restarted or shutdown remotely by an user connected as an administrator (userlevel 100), using the commands "system.server.restart" and "system.server.shutdown". The Nebulae Administrator tool includes these commands and is available at the download section of Tabuleiro web site. It can be used for server administration using any browser with the current Shockwave plugin.

If you need to stop a server you can also find its process id from the shell:

```
ps -auxww
```

This will show the process IDs for all running applications. Locate the Nebulae server process (it contains the java application name) and kill it with:

```
kill pidnumber
```

NEBULAE MUTIUSER SERVER HELP: INSTALLING THE SERVER

INSTALLING ON MACOSX

It is possible to install Nebulae in a MacOSX server installation using the same procedure detailed for Solaris and Linux installations. MacOS X contains one of the latest and fastest versions of Sun's Java 2 Virtual Machine, and it is an excellent platform for Nebulae MultiUser Server deployment and testing.

We also have a disk image file for MacOSX available at the download pages at xtras.tabuleiro.com. Double click the file to mount a disk named Nebulae MultiUser Server on the desktop. There are two separate directories: one contains a Nebulae subdiretory, which should be dragged to:

/Applications/Nebulae

This is also the location where the server log file and databases will be created. Before the server is run for the first time the directory also contains the files Nebulae.jar, hsqldb.jar, ScriptMap.cfg, Movie.cfg and Nebulae.cfg. If you have an existing installation, just copy the updated .jar files to it.

The second folder contains optional scripts that can be dragged to /Library/StartupItems/Nebulae. If installed these scripts will start up Nebulae automatically when the system is started. The StartupItems package installation is optional.

Before the server runs for the first time it is recommended that you create at least one administrative user in the server database, to allow access to server administration commands in the future. This can be accomplished by editing the "CreateUser" directive in the Nebulae.cfg file, created in the /Applications/Nebulae folder. Locate the line:

```
#CreateUser = admin,pass,100
```

Uncomment the line by removing the pound sign at the beginning, and customize the values for username and password. This command will create a user names "admin", with password "pass" and userlevel 100 the first time the server is started. Please note that while the server can operate without a server administrator account you will not be able to shutdown it remotely until an administrative user is added. You can now reboot your machine and the server will be started automatically by the StartupItem script.

It is also possible to start Nebulae from the a terminal window. This version of the command adds the Nebulae directory to the classpath, so you can put server side script classes in the same directory of Nebulae (/usr/local/nebulae) in order to have them loaded by your movies:

```
cd /Applications/Nebulae

unlimit

java -cp -Dfile.encoding=ISO-8859-1 Nebulae.jar:hsqldb.jar:/Applications/Nebulae
net.tabuleiro.nebulae.Nebulae &
```

For your convenience Nebulae can be restarted or shutdown remotely by an user connected as an administrator (userlevel 100), using the commands "system.server.restart" and "system.server.shutdown". The Nebulae Administrator tool includes these commands and is available at the download section of Tabuleiro web site. It can be used for server administration using any browser with the current Shockwave plugin.

If you need to stop a server you can also open a terminal window and find the server process id from the shell:

```
ps -auxww
```

This will show the process IDs for all running applications. Locate the Nebulae server process (it contains the java application name) and kill it with:

```
kill pidnumber
```

If you have installed the optional StartupItems package and do not want the server to start automatically in the future just remove the /Libraries/StartupItems/Nebulae directory. To remove the startup scripts you can type the following commands from a terminal window:

```
cd /Libraries/StartupItems

sudo rm -rf Nebulae
```

You will be asked for a root password for the system in order to remove the files.

NEBULAE MUTIUSER SERVER HELP: INSTALLING THE SERVER

INSTALLING ON WINDOWS

Most developers will use Macromedia's Shockwave MultiUser Server in Windows servers. However it is also possible to run Nebulae on Windows system, for deployment or testing purposes. We recommend using the latest Sun VM available at Sun's Java website. Windows 2000/XP is one of the most powerful platforms for running Java applications, and can handle a large number of users.

Nebulae is distributed in a .tar file available at the DOWNLOAD section on xtras.tabuleiro.com. Winzip can decompress tar files. You should decompress the archive to any directory in your system (C:\Nebulae for example.) This will create files Nebulae.jar, hsqldb.jar, ScriptMap.cfg, Nebulae.cfg and Movie.cfg. The original .tar file can be deleted at this time.

Before the server runs for the first time it is recommended that you create at least one administrative user in the server database, to allow access to server administration commands in the future. This can be accomplished by editing the "CreateUser" directive in the Nebulae.cfg file:

```
#CreateUser = admin,pass,100
```

Uncomment the line by removing the pound sign at the beginning, and customize the values for username and password. This command will create a user names "admin", with password "pass" and userlevel 100 the first time the server is started. Please note that while the server can operate without a server administrator account you will not be able to shutdown it remotely until an administrative user is added. You can use Notepad or other text editor to edit the configuration file.

From the command prompt you can issue the following command to verify that the Java VM installation is functional:

```
java -version
```

The result will be something like:

```
java version "1.6.0"

Java(TM) SE Runtime Environment (build 1.6.0-b105)

Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode, sharing)
```

You can start the Nebulae server with the following commands. Don't forget to add the Nebulae installation directory to the classpath if you want to load server side script classes, and put them in the same directory of Nebulae:

```
cd C:\Nebulae

java -cp Nebulae.jar;hsqldb.jar;C:\Nebulae net.tabuleiro.nebulae.Nebulae
```

Please remember to run the command from the directory where Nebulae is installed.

For your convenience Nebulae can be restarted or shutdown remotely by an user connected as an administrator (userlevel 100), using the commands "system.server.restart" and "system.server.shutdown". The Nebulae Administrator tool includes these commands and is available at the download section of Tabuleiro web site. It can be used for server administration using any browser with the current Shockwave plugin.

If you need to stop a server locally press CTR-C in the command prompt window where the server has started.

NEBULAE MUTIUSER SERVER HELP: INSTALLING THE SERVER

SAMPLE SERVER CONFIGURATION FILE

NEBULAE MUTIUSER SERVER HELP: INSTALLING THE SERVER

SAMPLE MOVIE CONFIGURATION FILE

NEBULAE MUTIUSER SERVER HELP: DATABASE SUPPORT

Nebulae servers support all the DBObject commands used in Shockwave MultiUser Server 2 and 3, and some new ones. These cover all DBAdmin, DBApplication, DBUser and DBPlayer functions. Older database commands used in SMUS 1 have been deprecated by Macromedia, and are NOT supported by Nebulae. A list of all commands and their syntax is available in the Command List section of this document.

The Nebulae download contains a package with HSQL Database engine classes . HSQLDB is free for redistribution and inclusion in commercial and non-commercial projects. All source code changes made by the Tabuleiro team to the HSQL engine were available to the HSQLDB developer community as part of the 1.6.1 release, and we participated in the development team for the project. HSQLDB is a very fast in-memory 100% Java database tool, and it runs in the same VM session of the Nebulae server for the best possible performance.

There is no configuration or installation needed to use databases in Nebulae. The server will automatically create and initialize a default database the first time it is started. The following files will be created in the server directory:

NebulaeDB.data

NebulaeDB.backup

NebulaeDB.script

NebulaeDB.properties

These files contain the cached version of the database. It is recommended that at least one administrative user is created in the database, to allow access to administrative database and server administration commands in the future. This can be accomplished with the "CreateUser" directive in the Nebulae.cfg file:

```
CreateUser = admin,pass,100
```

This command will create a user names "admin", with password "pass" and userlevel 100 the next time the server is started. We have authored a Director tool called "DatabaseAdministrator" that is available in the DOWNLOAD section of the Nebulae site to make it easier to manipulate and administrate the database contents.

But you can also use any other Director movie to interface with the database, as long as it uses the supported DBObject commands.

NEBULAE MUTIUSER SERVER HELP: SQL DATABASE SUPPORT

Nebulae servers can open a connection to standard database engines like Microsoft Access, Microsoft SQL Server, Oracle, MySQL and PostgreSQL, using the JDBC technology. This connection is established when the Nebulae server starts, following directives configured in the Nebulae.cfg. Users connected to a movie can use special commands to send SQL queries to the database. Commands are automatically translated by Nebulae and the results are returned as Lingo values, ready to be used in your Shockwave movies.

SQL commands available to Nebulae clients are:

System.SQL.executeQuery

System.SQL.executeUpdate

System.SQL.connect

System.SQL.disconnect

Please consult the Nebulae Command list section of the documentation for more information about the exact syntax for each command.

The technotes area at http://xtras.tabuleiro.com contains more information about connection to SQL databases, including updated configuration options for the most common database engines and instructions for setting up an external SQL database to be used for user authentication in the server.

NEBULAE MUTIUSER SERVER HELP: SERVER SIDE SCRIPTING

Tabuleiro Nebulae MultiUser Server version 1.1 introduced a powerful new feature: Java server side scripting. This opened Nebulae for deep customization, allowing the developer to extend the server in almost any way. Server side scripts can implement connection to databases, web servers, mail programs and almost any piece of software that can be accessed by a java application.

Server side scripts are an advanced feature in Nebulae. A good understanding of the Java language is absolutely required, and developers need access to a Java development environment in order to test and compile their server side script classes. Server side scripts are actually compiled Java .class files, mapped to Nebulae movies using the scriptmap.cfg file.

Scriptmap.cfg file configuration

A server side script class is loaded by Nebulae when a movie is created on the server and it matches the name of a movie configured in the scriptmap table. Each server side script is a compiled Java Class that extends the ServerSideScript class, defined in the Nebulae server side scripting API. The methods inherited from the ServerSideScript class allow the script to be called by Nebulae when a message with the subject "system.script.*" is received by the server. ServerSideScripts also implement the ServerUser interface, so they can join movies and groups and send messages as if they were an user connected to the Nebulae server.

The Nebulae server side scripting API is documented in javadoc format. Classes are provided for conversion between Java types and native Lingo values (LValues), and to encapsulate the Shockwave MultiUser Server message protocol. An effort has been made to provide interfaces that match the Server Object Model defined by Macromedia for SMUS 3. Java developers should probably start by examining the ServerObject, ServerMovie, ServerGroup and ServerUser interfaces.

Nebulae scripting API also exposes interfaces to the ServerUserDatabase and SQLGateway objects used by the server, so scripts can interact directly with databases and return results to connected users as native Lingo types, represented by LValues in the Nebulae API. Almost all aspects of the server can be accessed by a ServerSideScript class, including every movie, group or user connected.

The best way to learn the Nebulae Server Side Scripting API is to download some sample code and study the documentation provided. The technotes area at xtras.tabuleiro.com contains more information about server side scripting with Nebulae, including sample server side scripts that you can study, modify and use in your server.

DISCLAIMER: Tabuleiro technical support will be able to assist you with questions specific to the Nebulae API and overall design of the interfaces, but we can not write or debug Java code specifically for you. Please consult the SUPPORT POLICIES page at out web site for more information about our standard support policies.

NEBULAE MUTIUSER SERVER HELP: SERVER SIDE SCRIPTING

SAMPLE SCRIPTMAP CONFIGURATION FILE

NEBULAE MUTIUSER SERVER HELP: MUS COMMAND LIST

Nebulae is 100% compatible with the standard Shockwave MultiUser Server commands as documented in the SMUS version 3 manual, "Using the Shockwave MultiUser Server and Xtra". Almost all multiuser movies authored to SMUS 3 can connect to a Nebulae server seamlessly, with no modification required. Below is a list of all standard MUS commands supported by Nebulae. We recommend using the standard MUS command syntax described in the Director manuals to keep your movies compatible with future versions of Nebulae and the SMUS server, so please refer to your Director documentation for more information about movie syntax. Please note that the older database commands for SMUS version 1 are NOT supported.

system.server.getTime

system.server.getVersion

system.server.getMovieCount

system.server.getMovies

system.movie.getUserCount

system.movie.getGroups

system.movie.getGroupCount

system.movie.enable

system.movie.disable

system.movie.delete

system.group.join

system.group.leave

system.group.getUserCount

system.group.getUsers

system.group.enable

system.group.disable

system.group.delete

system.group.createUniqueName

system.group.setAttribute

system.group.getAttribute

system.group.getAttributeNames

system.group.deleteAttribute

system.user.changeMovie *

system.user.getGroups

system.user.getGroupCount

system.user.getAddress

system.user.delete

system.DBAdmin.createApplication

system.DBAdmin.deleteApplication

system.DBAdmin.declareAttribute

system.DBAdmin.deleteAttribute

system.DBAdmin.createApplicationData

system.DBAdmin.deleteApplicationData

system.DBAdmin.getUserCount **

system.DBAdmin.getUserNames **

system.DBApplication.setAttribute

system.DBApplication.getAttribute

system.DBApplication.getAttributeNames

system.DBApplication.deleteAttribute

system.DBApplication.getApplicationData

system.DBUser.setAttribute

system.DBUser.getAttribute

system.DBUser.getAttributeNames

system.DBUser.deleteAttribute

system.DBPlayer.setAttribute

system.DBPlayer.getAttribute

system.DBPlayer.getAttributeNames

system.DBPlayer.deleteAttribute

* changeMovie is only documented in the Release Notes file for the SMUS 3 server

** these commands are not listed in the command list summary for the SMUS 3 manual but appear in other sections of the document

NEBULAE MUTIUSER SERVER HELP: NEBULAE COMMAND LIST

Nebulae also implements extended commands not available in Macromedia's Shockwave MultiUser server. Most are used for server administration purposes, but there are a few extensions to the DBObject commands that were planned to make the life of the multiuser developer easier. These commands will not produce an error when used with the Shockwave MultiUser server but you should probably check the response to "system.server.getVersion" to make sure you are connected to a Nebulae server before using them.

NEBULAE SQL COMMANDS

system.SQL.executeQuery

Description: Executes an SQL query through Nebulae's SQL gateway. Nebulae constructs a prepared SQL statement using the query and Lingo values supplied, which are automatically mapped to the corresponding SQL data types. This avoids the problem of quoting strings, a common issue in all SQL queries. For simple queries (containing no ? elements) the #values parameter is an empty list.

Message recipient: system.SQL.executeQuery

Content parameters: [#sql:"sql query ",#values:[]]

Lingo example: sendNetMessage(system.SQL.executeQuery , anysubject , [#sql:"SELECT NAME,PHONE FROM ADDRESSES WHERE STREET=? AND NUMBER>?",#values:["California",1000]])

Server reply content: [["John Taylor","5551234"], ["Anthony Fields","5551432"]]

Remarks: Nebulae will return each matching row as a separate list inside a master list, with the values converted from SQL types to the appropriate Lingo types.

system.SQL.executeUpdate

Description: Executes an SQL update through Nebulae's SQL gateway. An SQL update is an instruction that does not return a selection (INSERT INTO, UPDATE, CREATE, etc.) Nebulae constructs a prepared SQL statement using the query and Lingo values supplied, which are automatically mapped to the corresponding SQL data types. This avoids the problem of quoting strings, a common issue in all SQL queries. For simple queries (containing no ? elements) the #values parameter is an empty list.

Message recipient: system.SQL.executeUpdate

Content parameters: [#sql:"sql query ",#values:[]]

Lingo example: sendNetMessage(system.SQL.executeUpdate , anysubject , [#sql:"UPDATE ADDRESSES SET PHONE=? WHERE NAME=?",["5551234","John Taylor"]])

Server reply content: "UpdateOK", or a MUS error message

Remarks: None

system.SQL.connect

Description: Connects Nebulae to an SQL database. This command is usually not necessary, since Nebulae connects automatically to the default SQL database when the server starts, using the SQL directive in the Nebulae.cfg file.

Each Nebulae server can only be connected to one SQL database at any given time, so this command will fail if a connection is already established. The parameters correspond to JDBC parameters for a database connection.

Message recipient: system.SQL.connect

Content parameters: [#userid:"name",#password:"",#driver:"org.hsqldb.jdbcDriver", #url:"jdbc:hsqldb:NebulaeSQL"]

Lingo example: sendNetMessage(system.SQL.connect , anysubject , [#userid:"sa",#password:"",#driver:"org.hsqldb.jdbcDriver", #url:"jdbc:hsqldb:NebulaeSQL"])

Server reply content: "Connected", or a MUS error message.

Remarks: By default this command can only be issued by users connected with administrative access level.

system.SQL.disconnect

Description: Disconnects Nebulae to an SQL database. This command is usually not necessary, since Nebulae manages the default SQL connection automatically.

Message recipient: system.SQL.disconnect

Content parameters: none

Lingo example: sendNetMessage(system.SQL.disconnect , anysubject , void)

Server reply content: void

Remarks: By default this command can only be issued by users connected with administrative access level.

NEBULAE EXTENSIONS TO THE SYSTEM.SERVER COMMAND SET

system.server.restart

Description: This command will restart the Nebulae MultiUser Server instance remotely. The actual java process running on the host machine will not be destroyed, however. When this command is received all users will be disconnected and a new server instance will be spawned. All configuration files will be re-scanned when the server restarts. The server restart process takes an average of 15 seconds to complete, and all database files are also packed and reopened.

Message recipient: system.server.restart

Content parameters: None

Lingo example: sendNetMessage(system.server.restart , anysubject ,void)

Server reply content: "ServerRestarted"

Remarks: By default this command can only be issued by users connected with administrative access level.

system.server.disable

Description: Prevents any new connections to the server. Current user connections will be unaffected. It is important to understand that even administrative users will NOT be able to connect to a disabled server, so if the administrative connection that issued the command is lost you will need to kill and restart the server process from the shell. This command can be reversed with the system.server.enable command, or with a server restart or reboot.

Message recipient: system.server.disable

Content parameters: None

Lingo example: sendNetMessage(system.server.disable , anysubject ,void)

Server reply content: "ServerDisabled"

Remarks: By default this command can only be issued by users connected with administrative access level.

system.server.enable

Description: Reverts the effects of system.server.disable, and allows new connections to the server.

Message recipient: system.server.enable

Content parameters: None

Lingo example: sendNetMessage(system.server.enable , anysubject ,void)

Server reply content: "ServerEnabled"

Remarks: By default this command can only be issued by users connected with administrative access level.

system.server.disconnectAll

Description: Disconnects all users from the server, including the administrative user that issued the command. This will cause all server movies and groups to be destroyed.

Message recipient: system.server.disconnectAll

Content parameters: None

Lingo example: sendNetMessage(system.server.disconnectAll , anysubject ,void)

Server reply content: "DisconnectAll"

Remarks: By default this command can only be issued by users connected with administrative access level.

system.server.sendEmail

Description: Sends an email using the Nebulae server and the specified SMTP host.

Message recipient: system.server.sendEmail

Content parameters: [#sender:"sender@domain.com", #recpt:"email@domain.com", #subject:"Message subject", #smtphost:"smtphostname.com", #data:["Line 1 of the message","Line 2 of the message","Line 3, etc"]]

Lingo example:
sendNetMessage(system.server.sendEmail , anysubject ,[#sender:"admin@yourdomain.com",

#recpt:"email@youruser.com", #subject:"Testing email", #smtphost:"yourdomain.com", #data:["This is a test message.","There is no need to reply"])

Server reply content: "EmailAccepted"

Remarks: None

NEBULAE EXTENSIONS TO THE SYSTEM.DBADMIN COMMAND SET

system.DBAdmin.ban

Description: Bans an ip address or userid from connecting to the server for a specific time period. This command does not disconnect the user if it is currently connected (see system.user.delete.) The banned entry is stored in the server database and is preserved even if the server is restarted or rebooted. This commands expects a property list in the content portion of the message with the attributes #user and #timeToBan. #user is a string representing an ip address or an username, and #timeToBan is the duration of the banned entry in the server, in seconds.

Message recipient: system.DBAdmin.ban

Content parameters: [#user:"ip address or username", #timeToBan: howManySeconds]

Lingo example:
sendNetMessage(system.DBAdmin.ban , anysubject ,[#user:"200.215.123.12", #timeToBan: 3600); sendNetMessage(system.DBAdmin.ban , anysubject ,[#user:"MrX", #timeToBan: 604800)

Server reply content: [#user:"200.215.123.12"]

Remarks: By default this command can only be issued by users connected with administrative access level.

system.DBAdmin.revokeBan

Description: Erases a banned entry from the server database. This commands expects a property list in the content portion of the message with the attribute #user, representing an ip address or an username.

Message recipient: system.DBAdmin.revokeBan

Content parameters: [#user:"ip address or username"]

Lingo example:
sendNetMessage(system.DBAdmin.revokeBan , anysubject ,[#user:"200.215.123.12");
sendNetMessage(system.DBAdmin.revokeBan , anysubject ,[#user:"MrX")

Server reply content: [#user:"200.215.123.12"]

Remarks: By default this command can only be issued by users connected with administrative access level.

system.DBAdmin.getBanned

Description: Retrieves a list of banned user entries in the server database, including the expiration date for the ban. The list is returned as a list of property lists, each item corresponding to one banned user entry.

Message recipient: system.DBAdmin.getBanned

Content parameters: None

Lingo example: sendNetMessage(system.DBAdmin.getBanned , anysubject ,void)

Server reply content: [[#user:"200.215.123.12",#expires:"2001/11/21 14:34:33"],[#user:"MrX",#expires:"2001/11/28 14:54:45"]]

Remarks: By default this command can only be issued by users connected with administrative access level. The banned table is only checked for expired entries when the server is started or restarted, for performance reasons. The list returned by this command may contain expired entries, but these will be expurged automatically if a banned user or ip address tries to connect to the server after the expiration period has passed.

system.DBAdmin.getApplicationNames

Description: Retrieves a list of application names for all DBApplication objects in the server database.

Message recipient: system.DBAdmin.getApplicationNames

Content parameters: None

Lingo example: sendNetMessage(system.DBAdmin.getApplicationNames , anysubject ,void)

Server reply content: ["lobby","seabattle"]

Remarks: By default this command can only be issued by users connected with administrative access level.

system.DBAdmin.getApplicationCount

Description: Retrieves the number of DBApplication objects in the server database.

Message recipient: system.DBAdmin.getApplicationCount

Content parameters: None

Lingo example: sendNetMessage(system.DBAdmin.getApplicationCount , anysubject ,void)

Server reply content: 3

Remarks: By default this command can only be issued by users connected with administrative access level.

system.DBAdmin.getAttributeNames

Description: Retrieves a list with the names of all attributes declared in the server database, including the default ones.

Message recipient: system.DBAdmin.getAttributeNames

Content parameters: None

Lingo example: sendNetMessage(system.DBAdmin.getAttributeNames , anysubject ,void)

Server reply content: ["CREATIONTIME","STATUS","USERLEVEL", "DESCRIPTION","PASSWORD","LASTUPDATETIME","LASTLOGINTIME"]

Remarks: By default this command can only be issued by users connected with administrative access level.

system.DBAdmin.getAttributeCount

Description: Retrieves the number of attributes declared in the server database, including the default ones.

Message recipient: system.DBAdmin.getAttributeCount

Content parameters: None

Lingo example: sendNetMessage(system.DBAdmin.getAttributeCount , anysubject ,void)

Server reply content: 7

Remarks: By default this command can only be issued by users connected with administrative access level.

system.DBAdmin.deleteApplicationData (#all format)

Description: This command is an addition to the standard system.DBAdmin.deleteApplicationData command, and it is included to make it easier for a developer to delete all ApplicationData objects created for a given application. The standard command supplied by Shockwave MultiUser Server requires a #text, #number or #range parameter that is used to match application data that is going to be deleted. Using the #all parameter a Nebulae server will be able to match and delete all ApplicationData objects created for the specified application.

Message recipient: system.DBAdmin.deleteApplicationData

Content parameters: [#application:"applicationname",#attribute:#creationtime, #all:void]

Lingo example: sendNetMessage(system.DBAdmin.deleteApplicationData , anysubject , [#application:"seabattle",#attribute:#creationtime, #all:void])

Server reply content: Matches the command message content

Remarks: By default this command can only be issued by users connected with administrative access level.

NEBULAE EXTENSIONS TO THE SYSTEM.DBAPPLICATION COMMAND SET

system.DBApplication.getApplicationData (#all format)

Description: This command is an addition to the standard system.DBApplication.getApplicationData command, and it is included to make it easier for a developer to retrieve all ApplicationData objects created for a given application. The standard command supplied by Shockwave MultiUser Server requires a #text, #number or #range parameter that is used to match application data that is going to be returned. Using the #all parameter a Nebulae server will be able to match and retrieve all ApplicationData objects created for the specified application.

Message recipient: system.DBApplication.getApplicationData

Content parameters: [#application:"applicationname",#attribute:#creationtime, #all:void]

Lingo example: sendNetMessage(system.DBApplication.getApplicationData , anysubject , [#application:"seabattle",#attribute:#creationtime, #all:void])

Server reply content: [[#shipname:"Cruise",#shipcolor:"Blue"], [#shipname:"Sub",#shipcolor:"Green"]]

Remarks: By default this command can only be issued by users connected with administrative access level.

NEBULAE MUTIUSER SERVER HELP: KNOWN ISSUES

It is important to understand that not all platforms and Java virtual machines (VM) are created equal. Some VM can support a larger number of concurrent user connections, while others are limited by the underlying operations system. Some platforms handle multiple threads faster, and scale better. The Nebulae server does not necessarily impose any limit on the number of simultaneous user connections or messages dispatched by seconds, but its abilities may be limited by the host VM and platform. It is up to the developer to test if a certain server configuration can scale to meet the requirements of a given application.

We recommend checking the SYSTEM REQUIREMENTS section at the Nebulae page on xtras.tabuleiro.com for updated information on the latest Java virtual machines tested with Nebulae and additional performance considerations.

NEBULAE MUTIUSER SERVER HELP: HOW TO ORDER AND REGISTER

The unregistered version of Tabuleiro Nebulae MultiUser Server may be used for evaluation, nonprofit and educational purposes only: commercial deployment is strictly prohibited. The trial version is limited to 10 simultaneous user connections. The registered version of Nebulae may be purchased online at xtras.tabuleiro.com using a secure server. At our web site you can also consult our purchase policy, purchase instructions, payment, delivery and security methods.

If you decide to buy the server you don't need to download a new copy of the software. After your order is processed you will receive an e-mail with a serial number and instructions to register the software you've already installed on your machine. Please keep your serial number archived, since it may be necessary if you move the software to a different machine.

NEBULAE MUTIUSER SERVER HELP: LICENSING AND AVAILABILITY

Tabuleiro Nebulae MultiUser Server is a commercial product. Current price and updated information can be found at xtras.tabuleiro.com.

A separate serial number is required for each machine or instance of the server running. Contact us if you need more info.

COPYRIGHT INFORMATION

NEBULAE MultiUser Server is a commercial product. A trial version is available to download for evaluation and it can be registered to unlock the connection limits of the program. TABULEIRO PROD. MAKES NO AND HEREBY SPECIFICALLY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, NON INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE TABULEIRO NEBULAE MultiUser Server. Tabuleiro does not warrant or guarantee that the software is error-free or that defects in the Software will be corrected. Tabuleiro is not responsible for any special, incidental, indirect or consequential damages. In no event will Tabuleiro's liability with respect to this license agreement exceed the amount you paid (if you paid) to Tabuleiro Prod. for the Software. IN NO EVENT WILL TABULEIRO PROD. BE LIABLE TO ANY PARTY FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS OR INFORMATION, AND THE LIKE), OR ANY OTHER DAMAGES ARISING IN ANY WAY OUT OF THE AVAILABILITY, USE, RELIANCE ON, OR INABILITY TO USE THE SOFTWARE.

Nebulae MultiUser Server Copyright (c) Tabuleiro Producoes Ltda, 2001-2003

This product includes Hypersonic SQL

Copyright (c) 1995-2000 by the Hypersonic SQL Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other

NEBULAE MUTIUSER SERVER HELP: TECHNICAL SUPPORT

Please use the Your Account section available at our web site xtras.tabuleiro.com to submit your questions. The site also contains Technotes and other resources that can help you identify and solve the most common problems quickly.

All Packages Class Hierarchy

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z _

Index of all Fields and Methods

A

addElement(LValue). Method in class net.tabuleiro.nebulae.LList
    Adds an LValue element to the list
addElement(LValue, LValue). Method in class net.tabuleiro.nebulae.LPropList
    Adds an LValue element to the list
addElement(MUSMsgHeaderString). Method in class net.tabuleiro.nebulae.MUSMsgHeaderStringList
    Adds a MUSMsgHeaderString to the list
addUser(ServerUser). Method in interface net.tabuleiro.nebulae.ServerGroup
    Adds an user to this server group
AttributeNotFoundException(String). Constructor for class net.tabuleiro.nebulae.AttributeNotFoundException

B

BadConnectionID. Static variable in class net.tabuleiro.nebulae.MUSErrorCode
BadParameter. Static variable in class net.tabuleiro.nebulae.MUSErrorCode
binHexToBytes(String, byte[], int, int, int). Static method in class net.tabuleiro.nebulae.ConversionUtils
byteArrayToInt(byte[], int). Static method in class net.tabuleiro.nebulae.ConversionUtils
byteArrayToLong(byte[], int). Static method in class net.tabuleiro.nebulae.ConversionUtils
byteArrayToShort(byte[], int). Static method in class net.tabuleiro.nebulae.ConversionUtils
byteArrayToUNCString(byte[], int, int). Static method in class net.tabuleiro.nebulae.ConversionUtils
bytesToBinHex(byte[]). Static method in class net.tabuleiro.nebulae.ConversionUtils
bytesToBinHex(byte[], int, int). Static method in class net.tabuleiro.nebulae.ConversionUtils

C

connect(String, String, String, String). Method in interface net.tabuleiro.nebulae.SQLGateway
    Connects to the default SQL database.
ConnectionDuplicate. Static variable in class net.tabuleiro.nebulae.MUSErrorCode
ConnectionRefused. Static variable in class net.tabuleiro.nebulae.MUSErrorCode
ConversionUtils(). Constructor for class net.tabuleiro.nebulae.ConversionUtils
correctLongBytes(long). Static method in class net.tabuleiro.nebulae.ConversionUtils
count(). Method in class net.tabuleiro.nebulae.LList
    Returns the number of elements in the list
count(). Method in class net.tabuleiro.nebulae.LPropList
    Returns the number of elements in the list
createServerGroup(String). Method in interface net.tabuleiro.nebulae.ServerMovie

Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LColor
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LDate
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LFloat
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LInteger
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LList
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LMedia
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LPicture
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LPoint
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LPropList
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LRect
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LString
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LSymbol
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.LValue
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.MUSMessage
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.MUSMsgHeaderString
Reserved for internal use of the Nebulae MultiUser Server.

dump(). Method in class net.tabuleiro.nebulae.MUSMsgHeaderStringList
Reserved for internal use of the Nebulae MultiUser Server.

E

elements(). Method in class net.tabuleiro.nebulae.MUSMsgHeaderStringList
Retrieves an Enumeration object containing the MUSMsgHeaderString elements.

enableGroup(String). Method in interface net.tabuleiro.nebulae.ServerMovie
Enables this group to receive new users.

ErrorJoiningGroup. Static variable in class net.tabuleiro.nebulae.MUSErrorCode

ErrorLeavingGroup. Static variable in class net.tabuleiro.nebulae.MUSErrorCode

executeQuery(String, LList). Method in interface net.tabuleiro.nebulae.SQLGateway
Executes an SQL query call
Use question marks in the query as placeholders for values contained in the params list.

executeUpdate(String, LList). Method in interface net.tabuleiro.nebulae.SQLGateway
Executes an SQL update call
Use question marks in the query as placeholders for values contained in the params list.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.L3dTransform

Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.L3dVector
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LColor
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LDate
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LFloat
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LInteger
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LList
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LMedia
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LPoint
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LPropList
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LRect
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LString
Reserved for internal use of the Nebulae MultiUser Server.

extractFromBytes(byte[], int). Method in class net.tabuleiro.nebulae.LValue
Reserved for internal use of the Nebulae MultiUser Server.

extractMUSMessage(byte[]). Method in class net.tabuleiro.nebulae.MUSMessage
Reserved for internal use of the Nebulae MultiUser Server.

extractMUSMsgHeaderString(byte[], int). Method in class net.tabuleiro.nebulae.MUSMsgHeaderString
Reserved for internal use of the Nebulae MultiUser Server.

extractMUSMsgHeaderStringList(byte[], int). Method in class net.tabuleiro.nebulae.MUSMsgHeaderStringList
Reserved for internal use of the Nebulae MultiUser Server.

F

fromRawBytes(byte[], int). Static method in class net.tabuleiro.nebulae.LValue
Static function to construct an LValue from a raw byte array containg a Lingo formatted value and associated type information.

G

getBytes(). Method in class net.tabuleiro.nebulae.L3dTransform
Reserved for internal use of the Nebulae MultiUser Server.

getBytes(). Method in class net.tabuleiro.nebulae.L3dVector
Reserved for internal use of the Nebulae MultiUser Server.

getBytes(). Method in class net.tabuleiro.nebulae.LColor
Reserved for internal use of the Nebulae MultiUser Server.

getBytes(). Method in class net.tabuleiro.nebulae.LDate

getLValue(double). Static method in class net.tabuleiro.nebulae.LValue
Static function to construct an LValue from a Java double.

getLValue(float). Static method in class net.tabuleiro.nebulae.LValue
Static function to construct an LValue from a Java float.

getLValue(int). Static method in class net.tabuleiro.nebulae.LValue
Static function to construct an LValue from a Java int.

getLValue(String). Static method in class net.tabuleiro.nebulae.LValue
Static function to construct an LValue from a Java String.

getPropAt(int). Method in class net.tabuleiro.nebulae.LPropList
Fetches an LValue property name from the list

getServerGroup(int). Method in interface net.tabuleiro.nebulae.ServerMovie
Retrieves a pointer to a ServerGroup object representing a group that exists on this ServerMovie.

getServerGroup(String). Method in interface net.tabuleiro.nebulae.ServerMovie
Retrieves a pointer to a ServerGroup object representing a group that exists on this ServerMovie.

getServerMovie(int). Method in interface net.tabuleiro.nebulae.ServerObject
Retrieves a pointer to a ServerMovie object active on the server.

getServerMovie(String). Method in interface net.tabuleiro.nebulae.ServerObject
Retrieves a pointer to a ServerMovie object active on the server.

getServerUser(int). Method in interface net.tabuleiro.nebulae.ServerGroup
Retrieves a pointer to a ServerUser object representing a user connected to the movie.

getServerUser(String). Method in interface net.tabuleiro.nebulae.ServerGroup
Retrieves a pointer to a ServerUser object representing a user connected to the movie.

getServerUserDatabase(). Method in interface net.tabuleiro.nebulae.ServerObject
Returns a ponter to the ServerUserDatabase interface representing the default user authentication table.

getSQLGateway(). Method in interface net.tabuleiro.nebulae.ServerObject
Returns a ponter to the SQLGateway interface representing the default SQL connection.

getType(). Method in class net.tabuleiro.nebulae.LValue
Returns the type of an LValue (LValue.vtVoid, LValue.vtString, etc.)

groupCreate(ServerGroup). Method in class net.tabuleiro.nebulae.ServerSideScript
Called by the Nebulae server when a group is created in the movie associated with this server side script.

groupDelete(ServerGroup). Method in class net.tabuleiro.nebulae.ServerSideScript
Called by the Nebulae server when a group is deleted from the movie associated with this server side script.

groupJoin(ServerUser, ServerGroup). Method in class net.tabuleiro.nebulae.ServerSideScript
Called by the Nebulae server when an user joins a group in the movie associated with this server side script.

groupJoined(ServerGroup). Method in class net.tabuleiro.nebulae.ServerSideScript
Called by the Nebulae server when the script joins a group.

groupJoined(ServerGroup). Method in interface net.tabuleiro.nebulae.ServerUser
Called by the Nebulae server when the user joins a group.

groupLeave(ServerUser, ServerGroup). Method in class net.tabuleiro.nebulae.ServerSideScript
Called by the Nebulae server when an user leaves a group in the movie associated with this server side script.

groupLeft(ServerGroup). Method in class net.tabuleiro.nebulae.ServerSideScript
Called by the Nebulae server when the script leaves a group.

groupLeft(ServerGroup). Method in interface net.tabuleiro.nebulae.ServerUser
Called by the Nebulae server when the user leaves a group.

GroupNotFoundException(String). Constructor for class net.tabuleiro.nebulae.GroupNotFoundException

Log(Exception, int). Static method in class net.tabuleiro.nebulae.MUSLog

Logs a java exception to output.

Log(String, int). Static method in class net.tabuleiro.nebulae.MUSLog

Log as message to output.

longHi32(long). Static method in class net.tabuleiro.nebulae.ConversionUtils

longLo32(long). Static method in class net.tabuleiro.nebulae.ConversionUtils

longToBinHex(long). Static method in class net.tabuleiro.nebulae.ConversionUtils

longToByteArray(long, byte[], int). Static method in class net.tabuleiro.nebulae.ConversionUtils

longToIntArray(long, int[], int). Static method in class net.tabuleiro.nebulae.ConversionUtils

LPicture(). Constructor for class net.tabuleiro.nebulae.LPicture

Constructor

LPicture(byte[]). Constructor for class net.tabuleiro.nebulae.LPicture

Constructor

LPoint(). Constructor for class net.tabuleiro.nebulae.LPoint

Constructor

LPoint(LValue, LValue). Constructor for class net.tabuleiro.nebulae.LPoint

Constructor.

LPropList(). Constructor for class net.tabuleiro.nebulae.LPropList

Constructor

LRect(). Constructor for class net.tabuleiro.nebulae.LRect

Constructor

LRect(LValue, LValue, LValue, LValue). Constructor for class net.tabuleiro.nebulae.LRect

Constructor.

LString(). Constructor for class net.tabuleiro.nebulae.LString

Constructor

LString(String). Constructor for class net.tabuleiro.nebulae.LString

Constructor

LSymbol(). Constructor for class net.tabuleiro.nebulae.LSymbol

Constructor.

LSymbol(String). Constructor for class net.tabuleiro.nebulae.LSymbol

Constructor.

LValue(). Constructor for class net.tabuleiro.nebulae.LValue

Constructor

LVoid(). Constructor for class net.tabuleiro.nebulae.LVoid

Constructor

M

m_args. Variable in class net.tabuleiro.nebulae.MUSDBMessage

m_dispatcher. Variable in class net.tabuleiro.nebulae.MUSMovie

Member variable not exposed to server side scripting

m_errCode. Variable in class net.tabuleiro.nebulae.MUSErrorCode

m_errCode. Variable in class net.tabuleiro.nebulae.MUSMessage

Message error code, represented as a MUSErrorCode type (for example MUSErrorCode.NoError)

m_floats. Variable in class net.tabuleiro.nebulae.L3dTransform

Array of 16 floats representing the Lingo Transform

m_floats. Variable in class net.tabuleiro.nebulae.L3dVector

Array of 3 floats representing the Lingo Vector

m_H. Variable in class net.tabuleiro.nebulae.LRect

H Coordinate of the Rect stored as an LValue.

m_header. Static variable in class net.tabuleiro.nebulae.MUSMessage
Default MUS message header, included automatically with each message.

m_list. Variable in class net.tabuleiro.nebulae.LList
Public vector element storing the list members as LValues

m_list. Variable in class net.tabuleiro.nebulae.LPropList
Public vector element storing the list members as LValues

m_LogLevel. Static variable in class net.tabuleiro.nebulae.MUSLog
Bitmaks value set automatically by Nebulae from the config file log directives

m_mov. Variable in class net.tabuleiro.nebulae.MUSDBMessage

m_msg. Variable in class net.tabuleiro.nebulae.MUSDBMessage

m_msgContent. Variable in class net.tabuleiro.nebulae.MUSMessage
The content part of this message.

m_name. Variable in class net.tabuleiro.nebulae.MUSGroup
Member variable not exposed to server side scripting

m_name. Variable in class net.tabuleiro.nebulae.ServerSideScript
The name of the script object.

m_proplist. Variable in class net.tabuleiro.nebulae.LPropList
Public vector element storing the property names as LSymbols

m_recptID. Variable in class net.tabuleiro.nebulae.MUSMessage
A MUSMsgHeaderStringList object containing one or more MUSMsgHeaderStrings, each corresponding to one intended recipient for this message.

m_reply. Variable in class net.tabuleiro.nebulae.MUSDBMessage

m_senderID. Variable in class net.tabuleiro.nebulae.MUSMessage
A single MUSMsgHeaderString object corresponding to the name of the message's sender.

m_stringlist. Variable in class net.tabuleiro.nebulae.MUSMsgHeaderStringList
Public vector element storing the MUSMsgHeaderString members.

m_subject. Variable in class net.tabuleiro.nebulae.MUSMessage
A single MUSMsgHeaderString object corresponding to the message's subject.

m_timeStamp. Variable in class net.tabuleiro.nebulae.MUSMessage
Message timestamp.

m_udp. Variable in class net.tabuleiro.nebulae.MUSMessage
UDP flag for this message.

m_user. Variable in class net.tabuleiro.nebulae.MUSDBMessage

m_userlevel. Variable in class net.tabuleiro.nebulae.ServerSideScript
The user access level for this script.

m_W. Variable in class net.tabuleiro.nebulae.LRect
W Coordinate of the Rect stored as an LValue.

m_X. Variable in class net.tabuleiro.nebulae.LPoint
X Coordinate of the Point stored as an LValue.

m_X. Variable in class net.tabuleiro.nebulae.LRect
X Coordinate of the Rect stored as an LValue.

m_Y. Variable in class net.tabuleiro.nebulae.LPoint
Y Coordinate of the Point stored as an LValue.

m_Y. Variable in class net.tabuleiro.nebulae.LRect
Y Coordinate of the Rect stored as an LValue.

main(String[]). Static method in class net.tabuleiro.nebulae.Nebulae
Entry point for the application

makeLong(int, int). Static method in class net.tabuleiro.nebulae.ConversionUtils

MessageContainsErrorInfo. Static variable in class net.tabuleiro.nebulae.MUSErrorCode

MessageMissing. Static variable in class net.tabuleiro.nebulae.MUSErrorCode

P

path(). Method in interface net.tabuleiro.nebulae.ServerObject
Gets the path of the server executable in the system
persists(). Method in interface net.tabuleiro.nebulae.ServerGroup
Checks if the group is set to persist on the movie even when no users members of it.
persists(). Method in interface net.tabuleiro.nebulae.ServerMovie
Checks if the movie is set to persist on the server even when no users are connected to it.
PlayerNotFoundException(String). Constructor for class net.tabuleiro.nebulae.PlayerNotFoundException
postMessage(MUSMessage). Method in class net.tabuleiro.nebulae.ServerSideScript
Posts a message to the Nebulae server dispatcher.
postMessage(MUSMessage). Method in interface net.tabuleiro.nebulae.ServerUser
Posts a message to the Nebulae server dispatcher.
PropertyNotFoundException(String). Constructor for class net.tabuleiro.nebulae.PropertyNotFoundException
put(String). Method in interface net.tabuleiro.nebulae.ServerObject
Displays a message to the server active output (log file or terminal window)

R

removeUser(ServerUser). Method in interface net.tabuleiro.nebulae.ServerGroup
Removes an user from this server group.
RequestedDataNotFound. Static variable in class net.tabuleiro.nebulae.MUSErrorCode
restart(). Static method in class net.tabuleiro.nebulae.Nebulae
Reserved for internal use of the Nebulae MultiUser Server

S

scriptCreate(). Method in class net.tabuleiro.nebulae.ServerSideScript
Called by the Nebulae server when the script object is created on the server.
scriptDelete(). Method in class net.tabuleiro.nebulae.ServerSideScript
Called by the Nebulae server when the script object is destroyed on the server.
sendMessage(MUSMessage). Method in interface net.tabuleiro.nebulae.ServerGroup
Sends a message to all users that are members of this group
sendMessage(MUSMessage). Method in class net.tabuleiro.nebulae.ServerSideScript
Sends a message to this script directly.
sendMessage(MUSMessage). Method in interface net.tabuleiro.nebulae.ServerUser
Sends a message to this server user directly.
ServerCloseFailed. Static variable in class net.tabuleiro.nebulae.MUSErrorCode
serverGroupCount(). Method in interface net.tabuleiro.nebulae.ServerMovie
Gets the number of groups in this movie.
ServerInitializationFailed. Static variable in class net.tabuleiro.nebulae.MUSErrorCode
ServerInternalError. Static variable in class net.tabuleiro.nebulae.MUSErrorCode
serverMovie(). Method in class net.tabuleiro.nebulae.ServerSideScript
Retrieves a pointer to the server movie object that created this script.

serverMovie(). Method in interface net.tabuleiro.nebulae.ServerUser

Retrieves the ServerMovie object representing the movie this user is connected to.

serverMovieCount(). Method in interface net.tabuleiro.nebulae.ServerObject

Gets the number of active movies in the server.

serverObject(). Method in class net.tabuleiro.nebulae.ServerSideScript

Retrieves a pointer to a ServerObject instance representing the current server.

ServerSendFailed. Static variable in class net.tabuleiro.nebulae.MUSErrorCode

ServerSideScript(). Constructor for class net.tabuleiro.nebulae.ServerSideScript

Constructor.

serverUserCount(). Method in interface net.tabuleiro.nebulae.ServerGroup

Gets the number of users in this group.

serverUserCount(). Method in interface net.tabuleiro.nebulae.ServerMovie

Gets the number of users connected to this movie.

setElementAt(int, LValue). Method in class net.tabuleiro.nebulae.LList

Stores an LValue element at the specified position

setLogLevel(int). Static method in class net.tabuleiro.nebulae.MUSLog

Integer value represent the log level bitmask

setpersists(boolean). Method in interface net.tabuleiro.nebulae.ServerGroup

Toggles the group persistent flag for this group.

setpersists(boolean). Method in interface net.tabuleiro.nebulae.ServerMovie

Toggles the movie persistent flag for this movie.

setType(short). Method in class net.tabuleiro.nebulae.LValue

Sets the type of a newly created LValue.

setuserLevel(int). Method in interface net.tabuleiro.nebulae.ServerMovie

Sets the default user levels for users that connect to this movie and do not have an user account in the DB.

setuserLevel(int). Method in interface net.tabuleiro.nebulae.ServerObject

Sets the default user levels for users that connect to the server and do not have an user account in the DB.

setuserLevel(int). Method in class net.tabuleiro.nebulae.ServerSideScript

Sets the user access level for this script.

setuserLevel(int). Method in interface net.tabuleiro.nebulae.ServerUser

Sets the user access level for this user.

setuserLimit(int). Method in interface net.tabuleiro.nebulae.ServerGroup

Sets the maximum number of users allowed to join this group.

shortToByteArray(int, byte[], int). Static method in class net.tabuleiro.nebulae.ConversionUtils

swapIntBytes(int). Static method in class net.tabuleiro.nebulae.ConversionUtils

T

timeStamp(). Method in interface net.tabuleiro.nebulae.ServerObject

Gets the current server time as an int

timeString(). Method in interface net.tabuleiro.nebulae.ServerObject

Gets the current server time formatted as a String

toBytes(). Method in class net.tabuleiro.nebulae.LColor

Returns the byte array storing the color data in binary format

toBytes(). Method in class net.tabuleiro.nebulae.LDate

Returns the byte array storing the date value in binary format

toBytes(). Method in class net.tabuleiro.nebulae.LMedia

Class Hierarchy

- class net.tabuleiro.nebulae.ConversionUtils
- class net.tabuleiro.nebulae.LValue
    - class net.tabuleiro.nebulae.L3dTransform
    - class net.tabuleiro.nebulae.L3dVector
    - class net.tabuleiro.nebulae.LColor
    - class net.tabuleiro.nebulae.LDate
    - class net.tabuleiro.nebulae.LFloat
    - class net.tabuleiro.nebulae.LInteger
    - class net.tabuleiro.nebulae.LList
    - class net.tabuleiro.nebulae.LMedia
        - class net.tabuleiro.nebulae.LPicture
    - class net.tabuleiro.nebulae.LPoint
    - class net.tabuleiro.nebulae.LPropList
    - class net.tabuleiro.nebulae.LRect
    - class net.tabuleiro.nebulae.LString
        - class net.tabuleiro.nebulae.LSymbol
    - class net.tabuleiro.nebulae.LVoid
- class net.tabuleiro.nebulae.MUSBlowfish
- class net.tabuleiro.nebulae.MUSDBMessage
- class net.tabuleiro.nebulae.MUSDispatcher
- class net.tabuleiro.nebulae.MUSGroup
- class net.tabuleiro.nebulae.MUSLog
- class net.tabuleiro.nebulae.MUSMessage
- class net.tabuleiro.nebulae.MUSMovie
- class net.tabuleiro.nebulae.MUSMsgHeaderString
- class net.tabuleiro.nebulae.MUSMsgHeaderStringList
- class net.tabuleiro.nebulae.MUSServer
- class net.tabuleiro.nebulae.Nebulae
- interface net.tabuleiro.nebulae.SQLGateway
- interface net.tabuleiro.nebulae.ServerGroup
- interface net.tabuleiro.nebulae.ServerMovie
- interface net.tabuleiro.nebulae.ServerObject
- class net.tabuleiro.nebulae.ServerSideScript (implements net.tabuleiro.nebulae.ServerUser)
- interface net.tabuleiro.nebulae.ServerUser
- interface net.tabuleiro.nebulae.ServerUserDatabase

API User's Guide Class Hierarchy Index

Package Index

Other Packages

- package net.tabuleiro.nebulae

All Packages Class Hierarchy Index

package net.tabuleiro.nebulae

Interface Index

- net.tabuleiro.nebulae.SQLGateway
- net.tabuleiro.nebulae.ServerGroup
- net.tabuleiro.nebulae.ServerMovie
- net.tabuleiro.nebulae.ServerObject
- net.tabuleiro.nebulae.ServerUser
- net.tabuleiro.nebulae.ServerUserDatabase

Class Index

- net.tabuleiro.nebulae.ConversionUtils
- net.tabuleiro.nebulae.L3dTransform
- net.tabuleiro.nebulae.L3dVector
- net.tabuleiro.nebulae.LColor
- net.tabuleiro.nebulae.LDate
- net.tabuleiro.nebulae.LFloat
- net.tabuleiro.nebulae.LInteger
- net.tabuleiro.nebulae.LList
- net.tabuleiro.nebulae.LMedia
- net.tabuleiro.nebulae.LPicture
- net.tabuleiro.nebulae.LPoint
- net.tabuleiro.nebulae.LPropList
- net.tabuleiro.nebulae.LRect
- net.tabuleiro.nebulae.LString
- net.tabuleiro.nebulae.LSymbol
- net.tabuleiro.nebulae.LValue
- net.tabuleiro.nebulae.LVoid
- net.tabuleiro.nebulae.MUSBlowfish
- net.tabuleiro.nebulae.MUSDBMessage
- net.tabuleiro.nebulae.MUSDispatcher
- net.tabuleiro.nebulae.MUSGroup
- net.tabuleiro.nebulae.MUSLog
- net.tabuleiro.nebulae.MUSMessage
- net.tabuleiro.nebulae.MUSMovie
- net.tabuleiro.nebulae.MUSMsgHeaderString
- net.tabuleiro.nebulae.MUSMsgHeaderStringList
- net.tabuleiro.nebulae.MUSServer
- net.tabuleiro.nebulae.Nebulae
- net.tabuleiro.nebulae.ServerSideScript

Exception Index

- net.tabuleiro.nebulae.AttributeNotFoundException
- net.tabuleiro.nebulae.DBException
- net.tabuleiro.nebulae.GroupNotFoundException
- net.tabuleiro.nebulae.MUSErrorCode
- net.tabuleiro.nebulae.MovieNotFoundException
- net.tabuleiro.nebulae.PlayerNotFoundException
- net.tabuleiro.nebulae.PropertyNotFoundException
- net.tabuleiro.nebulae.UserNotFoundException

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.AttributeNotFoundException

```
net.tabuleiro.nebulae.AttributeNotFoundException
```

public class **AttributeNotFoundException**
extends java.lang.Exception
implements java.io.Serializable

Constructor Index

- **AttributeNotFoundException**(String)

Constructors

AttributeNotFoundException

```
 public AttributeNotFoundException(java.lang.String msg)
```

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.ConversionUtils

```
net.tabuleiro.nebulae.ConversionUtils
```

public class **ConversionUtils**
extends java.lang.Object

Collection of static methods for data conversion. Data is treated in network byte order. This class is reserved for internal use of the Nebulae MultiUser Server

Constructor Index

- **ConversionUtils**()

Method Index

- **binHexToBytes**(String, byte[], int, int, int)
- **byteArrayToInt**(byte[], int)
- **byteArrayToLong**(byte[], int)
- **byteArrayToShort**(byte[], int)
- **byteArrayToUNCString**(byte[], int, int)
- **bytesToBinHex**(byte[])
- **bytesToBinHex**(byte[], int, int)
- **correctLongBytes**(long)
- **intArrayToLong**(int[], int)
- **intToBinHex**(int)
- **intToByteArray**(int, byte[], int)
- **longHi32**(long)
- **longLo32**(long)
- **longToBinHex**(long)
- **longToByteArray**(long, byte[], int)
- **longToIntArray**(long, int[], int)
- **makeLong**(int, int)
- **shortToByteArray**(int, byte[], int)
- **swapIntBytes**(int)

Constructors

ConversionUtils

```
 public ConversionUtils()
```

Methods

byteArrayToLong

```
public static long byteArrayToLong(byte buffer[],
                                   int nStartIndex)
```

byteArrayToInt

```
public static int byteArrayToInt(byte buffer[],
                                 int nStartIndex)
```

byteArrayToShort

```
public static short byteArrayToShort(byte buffer[],
                                     int nStartIndex)
```

longToByteArray

```
public static void longToByteArray(long lValue,
                                   byte buffer[],
                                   int nStartIndex)
```

correctLongBytes

```
public static long correctLongBytes(long lValue)
```

swapIntBytes

```
public static int swapIntBytes(int lValue)
```

intToByteArray

```
public static void intToByteArray(int lValue,
                                  byte buffer[],
                                  int nStartIndex)
```

shortToByteArray

```
public static void shortToByteArray(int lValue,
                                    byte buffer[],
                                    int nStartIndex)
```

intArrayToLong

```
public static long intArrayToLong(int buffer[],
                                  int nStartIndex)
```

longToIntArray

```
public static void longToIntArray(long lValue,
                                  int buffer[],
                                  int nStartIndex)
```

makeLong

```
public static long makeLong(int nLo,
                            int nHi)
```

longLo32

```
public static int longLo32(long lVal)
```

longHi32

```
public static int longHi32(long lVal)
```

bytesToBinHex

```
public static java.lang.String bytesToBinHex(byte data[])
```

longToBinHex

```
public static java.lang.String longToBinHex(long lValue)
```

intToBinHex

```
public static java.lang.String intToBinHex(int lValue)
```

bytesToBinHex

```
public static java.lang.String bytesToBinHex(byte data[],
                                             int nStartPos,
                                             int nNumOfBytes)
```

binHexToBytes

```
public static int binHexToBytes(java.lang.String sBinHex,
                                byte data[],
                                int nSrcPos,
                                int nDstPos,
                                int nNumOfBytes)
```

byteArrayToUNCString

```
public static java.lang.String byteArrayToUNCString(byte data[],
                                                    int nStartPos,
                                                    int nNumOfBytes)
```

All Packages Class Hierarchy This Package Previous Next Index

All Packages   Class Hierarchy   This Package   Previous   Next   Index

Class net.tabuleiro.nebulae.DBException

```
net.tabuleiro.nebulae.DBException
```

public class **DBException**
extends java.lang.Exception
implements java.io.Serializable

Constructor Index

- **DBException**(String)

Constructors

● DBException

```
 public DBException(java.lang.String msg)
```

All Packages   Class Hierarchy   This Package   Previous   Next   Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.GroupNotFoundException

```
net.tabuleiro.nebulae.GroupNotFoundException
```

public class **GroupNotFoundException**
extends java.lang.Exception
implements java.io.Serializable

Constructor Index

- **GroupNotFoundException**(String)

Constructors

- **GroupNotFoundException**

```
 public GroupNotFoundException(java.lang.String msg)
```

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.L3dTransform

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.L3dTransform
```

public class **L3dTransform**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible 3dTransform value (L3dTransform for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Variable Index

- **m_floats**
  Array of 16 floats representing the Lingo Transform

Constructor Index

- **L3dTransform**()
  Constructor
- **L3dTransform**(float, float, float, float, float, float, float, float, float, float, float, float, float, float, float, float)
  Constructor.

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.

Variables

m_floats

```
 public float m_floats[]
```

Array of 16 floats representing the Lingo Transform

Constructors

L3dTransform

```
public L3dTransform()
```

Constructor

L3dTransform

```
public L3dTransform(float a,
                    float b,
                    float c,
                    float d,
                    float e,
                    float f,
                    float g,
                    float h,
                    float i,
                    float j,
                    float k,
                    float l,
                    float m,
                    float n,
                    float o,
                    float p)
```

Constructor. 16 float values represent the Lingo Transform

Methods

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

extractFromBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LValue

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.L3dVector

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.L3dVector
```

public class **L3dVector**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible Vector value (L3dVector for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Variable Index

- **m_floats**
  Array of 3 floats representing the Lingo Vector

Constructor Index

- **L3dVector**()
  Constructor
- **L3dVector**(float, float, float)
  Constructor.

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.

Variables

m_floats

```
public float m_floats[]
```

Array of 3 floats representing the Lingo Vector

Constructors

L3dVector

```
public L3dVector()
```

Constructor

L3dVector

```
public L3dVector(float x,
                 float y,
                 float z)
```

Constructor. 3 float values represent the Lingo Vector (x,y,z)

Methods

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.
Overrides:
extractFromBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.
Overrides:
dump in class net.tabuleiro.nebulae.LValue

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.
Overrides:
getBytes in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LColor

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LColor
```

public class **LColor**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible color value (LColor for short) Color values are stored and retrieved as an opaque array of bytes Lingo is a trademark of Macromedia, Inc. All rights reserved.

Constructor Index

- **LColor**()
  Constructor
- **LColor**(byte[])
  Constructor

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **toBytes**()
  Returns the byte array storing the color data in binary format

Constructors

LColor

```
public LColor(byte initbytes[])
```

Constructor

LColor

```
public LColor()
```

Constructor

Methods

● toBytes

```
public byte[] toBytes()
```

Returns the byte array storing the color data in binary format

Overrides:

toBytes in class net.tabuleiro.nebulae.LValue

● extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

extractFromBytes in class net.tabuleiro.nebulae.LValue

● getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

● dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LDate

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LDate
```

public class **LDate**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible date value (LDate for short) Date values are stored and retrieved as an opaque array of bytes Lingo is a trademark of Macromedia, Inc. All rights reserved.

Constructor Index

- **LDate**()
  Constructor
- **LDate**(byte[])
  Constructor

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **toBytes**()
  Returns the byte array storing the date value in binary format

Constructors

LDate

```
 public LDate(byte initbytes[])
```

Constructor

LDate

```
 public LDate()
```

Constructor

Methods

toBytes

```
public byte[] toBytes()
```

Returns the byte array storing the date value in binary format

Overrides:

toBytes in class net.tabuleiro.nebulae.LValue

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

extractFromBytes in class net.tabuleiro.nebulae.LValue

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LFloat

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LFloat
```

public class **LFloat**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible Float value (LFloat for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Constructor Index

- **LFloat**()
  Constructor
- **LFloat**(double)
  Constructor

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **toDouble**()
  Returns this LFloat as a Java double.

Constructors

LFloat

```
public LFloat(double initInt)
```

Constructor

LFloat

```
public LFloat()
```

Constructor

Methods

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

extractFromBytes in class net.tabuleiro.nebulae.LValue

toDouble

```
public double toDouble()
```

Returns this LFloat as a Java double.

Overrides:

toDouble in class net.tabuleiro.nebulae.LValue

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LInteger

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LInteger
```

public class **LInteger**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible integer value (LInteger for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Constructor Index

- **LInteger**()
  Constructor
- **LInteger**(int)
  Constructor

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **toInteger**()
  Returns this LInteger value as an int.

Constructors

LInteger

```
public LInteger(int initInt)
```

Constructor

LInteger

```
public LInteger()
```

Constructor

Methods

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

extractFromBytes in class net.tabuleiro.nebulae.LValue

toInteger

```
public int toInteger()
```

Returns this LInteger value as an int.

Overrides:

toInteger in class net.tabuleiro.nebulae.LValue

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LList

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LList
```

public class **LList**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible List value (LList for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Variable Index

- **m_list**
  Public vector element storing the list members as LValues

Constructor Index

- **LList**()
  Constructor

Method Index

- **addElement**(LValue)
  Adds an LValue element to the list
- **count**()
  Returns the number of elements in the list
- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **getElementAt**(int)
  Fetches an LValue element from the list
- **setElementAt**(int, LValue)
  Stores an LValue element at the specified position

Variables

m_list

```
public java.util.Vector m_list
```

Public vector element storing the list members as LValues

Constructors

LList

```
public LList()
```

Constructor

Methods

addElement

```
public boolean addElement(net.tabuleiro.nebulae.LValue elem)
```

Adds an LValue element to the list
Parameters:
elem - LValue to add
Returns:
boolean

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.
Overrides:
extractFromBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.
Overrides:
dump in class net.tabuleiro.nebulae.LValue

getElementAt

```
public net.tabuleiro.nebulae.LValue getElementAt(int pos)
```

Fetches an LValue element from the list

Parameters:

pos - index of the element to be retrieved

Returns:

LValue

● setElementAt

```
public boolean setElementAt(int pos,
                            net.tabuleiro.nebulae.LValue elem)
```

Stores an LValue element at the specified position

Parameters:

pos - position to store the element

elem - LValue to store

Returns:

boolean

● count

```
public int count()
```

Returns the number of elements in the list

● getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LMedia

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LMedia
```

public class **LMedia**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible media value (LMedia for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Constructor Index

- **LMedia**()
  Constructor
- **LMedia**(byte[])
  Constructor

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **toBytes**()
  Returns the byte array storing the media data in binary format

Constructors

LMedia

```
public LMedia(byte initbytes[])
```

Constructor

LMedia

```
public LMedia()
```

Constructor

Methods

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

extractFromBytes in class net.tabuleiro.nebulae.LValue

toBytes

```
public byte[] toBytes()
```

Returns the byte array storing the media data in binary format

Overrides:

toBytes in class net.tabuleiro.nebulae.LValue

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LPicture

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LMedia
           |
           +----net.tabuleiro.nebulae.LPicture
```

public class **LPicture**
extends net.tabuleiro.nebulae.LMedia

Class representing a Lingo compatible picture value (LPicture for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Constructor Index

- **LPicture**()
  Constructor
- **LPicture**(byte[])
  Constructor

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.

Constructors

LPicture

```
public LPicture(byte initImage[])
```

Constructor

LPicture

```
public LPicture()
```

Constructor

Methods

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LMedia

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LPoint

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LPoint
```

public class **LPoint**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible Point value (LPoint for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Variable Index

- **m_X**
  X Coordinate of the Point stored as an LValue.
- **m_Y**
  Y Coordinate of the Point stored as an LValue.

Constructor Index

- **LPoint**()
  Constructor
- **LPoint**(LValue, LValue)
  Constructor.

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.

Variables

m_X

```
public net.tabuleiro.nebulae.LValue m_X
```

X Coordinate of the Point stored as an LValue. It can be an LInteger or an LFloat

m_Y

```
public net.tabuleiro.nebulae.LValue m_Y
```

Y Coordinate of the Point stored as an LValue. It can be an LInteger or an LFloat

Constructors

LPoint

```
public LPoint()
```

Constructor

LPoint

```
public LPoint(net.tabuleiro.nebulae.LValue x,
              net.tabuleiro.nebulae.LValue y)
```

Constructor. 2 LValues representing the Lingo Point coordinates

Methods

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.
Overrides:
extractFromBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.
Overrides:
dump in class net.tabuleiro.nebulae.LValue

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.
Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LPropList

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LPropList
```

public class **LPropList**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible List value (LList for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Variable Index

- **m_list**
  Public vector element storing the list members as LValues
- **m_proplist**
  Public vector element storing the property names as LSymbols

Constructor Index

- **LPropList**()
  Constructor

Method Index

- **addElement**(LValue, LValue)
  Adds an LValue element to the list
- **count**()
  Returns the number of elements in the list
- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **getElement**(LSymbol)
  Fetches an LValue element from the list
- **getElementAt**(int)
  Fetches an LValue element from the list
- **getPropAt**(int)
  Fetches an LValue property name from the list

Variables

m_proplist

```
public java.util.Vector m_proplist
```

Public vector element storing the property names as LSymbols

m_list

```
public java.util.Vector m_list
```

Public vector element storing the list members as LValues

Constructors

LPropList

```
public LPropList()
```

Constructor

Methods

addElement

```
public boolean addElement(net.tabuleiro.nebulae.LValue property,
                          net.tabuleiro.nebulae.LValue elem)
```

Adds an LValue element to the list

Parameters:

property - LSymbol with property name

elem - LValue to add

Returns:

boolean

getElementAt

```
public net.tabuleiro.nebulae.LValue getElementAt(int pos)
```

Fetches an LValue element from the list

Parameters:

pos - index of the element to be retrieved

Returns:

LValue

getPropAt

```
public net.tabuleiro.nebulae.LValue getPropAt(int pos)
```

Fetches an LValue property name from the list

Parameters:

pos - index of the property to be retrieved

Returns:

LValue

getElement

```
public synchronized net.tabuleiro.nebulae.LValue getElement(net.tabuleiro.nebulae.LSymbol prop
```

Fetches an LValue element from the list

Parameters:

prop - LSymbol representing the property name

Returns:

LValue

count

```
public int count()
```

Returns the number of elements in the list

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

extractFromBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LValue

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LRect

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LRect
```

public class **LRect**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible Rect value (LRect for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Variable Index

- **m_H**
  H Coordinate of the Rect stored as an LValue.
- **m_W**
  W Coordinate of the Rect stored as an LValue.
- **m_X**
  X Coordinate of the Rect stored as an LValue.
- **m_Y**
  Y Coordinate of the Rect stored as an LValue.

Constructor Index

- **LRect**()
  Constructor
- **LRect**(LValue, LValue, LValue, LValue)
  Constructor.

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.

Variables

m_X

```
public net.tabuleiro.nebulae.LValue m_X
```

X Coordinate of the Rect stored as an LValue. It can be an LInteger or an LFloat

m_Y

```
public net.tabuleiro.nebulae.LValue m_Y
```

Y Coordinate of the Rect stored as an LValue. It can be an LInteger or an LFloat

m_W

```
public net.tabuleiro.nebulae.LValue m_W
```

W Coordinate of the Rect stored as an LValue. It can be an LInteger or an LFloat

m_H

```
public net.tabuleiro.nebulae.LValue m_H
```

H Coordinate of the Rect stored as an LValue. It can be an LInteger or an LFloat

Constructors

LRect

```
public LRect()
```

Constructor

LRect

```
public LRect(net.tabuleiro.nebulae.LValue x,
             net.tabuleiro.nebulae.LValue y,
             net.tabuleiro.nebulae.LValue w,
             net.tabuleiro.nebulae.LValue h)
```

Constructor. 4 LValues representing the Lingo Rect coordinates

Methods

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

extractFromBytes in class net.tabuleiro.nebulae.LValue

● dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LValue

● getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LString

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LString
```

public class **LString**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible String value (LString for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Constructor Index

- **LString**()
  Constructor
- **LString**(String)
  Constructor

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **toString**()
  Returns this LString as a Java String.

Constructors

LString

```
public LString(java.lang.String initString)
```

Constructor

LString

```
public LString()
```

Constructor

Methods

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

extractFromBytes in class net.tabuleiro.nebulae.LValue

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

getBytes in class net.tabuleiro.nebulae.LValue

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LValue

toString

```
public java.lang.String toString()
```

Returns this LString as a Java String.

Overrides:

toString in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LSymbol

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LString
           |
           +----net.tabuleiro.nebulae.LSymbol
```

public class **LSymbol**
extends net.tabuleiro.nebulae.LString

Class representing a Lingo compatible Symbol value (LSymbol for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Constructor Index

- **LSymbol**()
  Constructor.
- **LSymbol**(String)
  Constructor.

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.

Constructors

LSymbol

```
 public LSymbol(java.lang.String initString)
```

Constructor. Calls superclass (LString) methods

LSymbol

```
 public LSymbol()
```

Constructor. Calls superclass (LString) methods

Methods

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

Overrides:

dump in class net.tabuleiro.nebulae.LString

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LValue

```
net.tabuleiro.nebulae.LValue
```

public class **LValue**
extends java.lang.Object

Base class representing a Lingo compatible value (LValue for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Variable Index

- **vt_3dTransform**
- **vt_3dVector**
- **vt_Color**
- **vt_Date**
- **vt_Float**
- **vt_Integer**
- **vt_List**
- **vt_Media**
- **vt_Picture**
- **vt_Point**
- **vt_PropList**
- **vt_Rect**
- **vt_String**
- **vt_Symbol**
- **vt_Void**

Constructor Index

- **LValue**()
  Constructor

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractFromBytes**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **fromRawBytes**(byte[], int)
  Static function to construct an LValue from a raw byte array containg a Lingo formatted value and associated type information.
- **getBytes**()

Reserved for internal use of the Nebulae MultiUser Server.

getLValue(byte[])
Static function to construct an LValue from an array of bytes.

getLValue(double)
Static function to construct an LValue from a Java double.

getLValue(float)
Static function to construct an LValue from a Java float.

getLValue(int)
Static function to construct an LValue from a Java int.

getLValue(String)
Static function to construct an LValue from a Java String.

getType()
Returns the type of an LValue (LValue.vtVoid, LValue.vtString, etc.)

setType(short)
Sets the type of a newly created LValue.

toBytes()
Base method, returns this LValue as a String.

toDouble()
Base method, returns this LValue as a String.

toInteger()
Base method, returns this LValue as a String.

toString()
Base method, returns this LValue as a String.

Variables

vt_Void

```
public static final short vt_Void
```

vt_Integer

```
public static final short vt_Integer
```

vt_Symbol

```
public static final short vt_Symbol
```

vt_String

```
public static final short vt_String
```

vt_Picture

```
public static final short vt_Picture
```

vt_Float

```
public static final short vt_Float
```

vt_List

```
public static final short vt_List
```

vt_Point

```
public static final short vt_Point
```

vt_Rect

```
public static final short vt_Rect
```

vt_PropList

```
public static final short vt_PropList
```

vt_Color

```
public static final short vt_Color
```

vt_Date

```
public static final short vt_Date
```

vt_Media

```
public static final short vt_Media
```

vt_3dVector

```
public static final short vt_3dVector
```

vt_3dTransform

```
public static final short vt_3dTransform
```

Constructors

LValue

```
public LValue()
```

Constructor

Methods

getLValue

```
public static net.tabuleiro.nebulae.LValue getLValue(int initval)
```

Static function to construct an LValue from a Java int. Returns an LInteger value

getLValue

```
public static net.tabuleiro.nebulae.LValue getLValue(java.lang.String initval)
```

Static function to construct an LValue from a Java String. Returns an LString value

getLValue

```
public static net.tabuleiro.nebulae.LValue getLValue(double initval)
```

Static function to construct an LValue from a Java double. Returns an LFloat value

getLValue

```
public static net.tabuleiro.nebulae.LValue getLValue(float initval)
```

Static function to construct an LValue from a Java float. Returns an LFloat value

getLValue

```
public static net.tabuleiro.nebulae.LValue getLValue(byte initval[])
```

Static function to construct an LValue from an array of bytes. Returns an LMedia value

setType

```
public void setType(short type)
```

Sets the type of a newly created LValue. Type is a short LValue type (LValue.vtVoid, LValue.vtString, etc.)

getType

```
public short getType()
```

Returns the type of an LValue (LValue.vtVoid, LValue.vtString, etc.)

extractFromBytes

```
public int extractFromBytes(byte rawBytes[],
                            int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

toString

```
public java.lang.String toString()
```

Base method, returns this LValue as a String. Not guaranteed to be implemented for all value types.

toInteger

```
public int toInteger()
```

Base method, returns this LValue as a String. Not guaranteed to be implemented for all value types.

toDouble

```
public double toDouble()
```

Base method, returns this LValue as a String. Not guaranteed to be implemented for all value types.

toBytes

```
public byte[] toBytes()
```

Base method, returns this LValue as a String. Not guaranteed to be implemented for all value types.

fromRawBytes

```
public static net.tabuleiro.nebulae.LValue fromRawBytes(byte rawBytes[],
                                                         int offset)
```

Static function to construct an LValue from a raw byte array containg a Lingo formatted value and associated type information. Reserved for internal use of the Nebulae MultiUser Server.

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.LVoid

```
net.tabuleiro.nebulae.LValue
   |
   +----net.tabuleiro.nebulae.LVoid
```

public class **LVoid**
extends net.tabuleiro.nebulae.LValue

Class representing a Lingo compatible Void value (LVoid for short). Lingo is a trademark of Macromedia, Inc. All rights reserved.

Constructor Index

- **LVoid**()
  Constructor

Method Index

- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.

Constructors

LVoid

```
 public LVoid()
```

Constructor

Methods

getBytes

```
 public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.
Overrides:
getBytes in class net.tabuleiro.nebulae.LValue

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MovieNotFoundException

```
net.tabuleiro.nebulae.MovieNotFoundException
```

public class **MovieNotFoundException**
extends java.lang.Exception
implements java.io.Serializable

Constructor Index

- **MovieNotFoundException**(String)

Constructors

MovieNotFoundException

```
 public MovieNotFoundException(java.lang.String msg)
```

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSBlowfish

```
net.tabuleiro.nebulae.MUSBlowfish
```

public class **MUSBlowfish**
extends java.lang.Object

This class is reserved for internal use of the Nebulae MultiUser Server

Constructor Index

- **MUSBlowfish**()

Method Index

- **decode**(String, byte[])
  Method not exposed to server side scripting

Constructors

● MUSBlowfish

```
 public MUSBlowfish()
```

Methods

● decode

```
 public static void decode(java.lang.String decodepattern,
                           byte data[])
```

Method not exposed to server side scripting

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSDBMessage

```
net.tabuleiro.nebulae.MUSDBMessage
```

public class **MUSDBMessage**
extends java.lang.Object

Base class representing a queued message to the database dispatcher.

Variable Index

- **m_args**
- **m_mov**
- **m_msg**
- **m_reply**
- **m_user**

Constructor Index

- **MUSDBMessage**(ServerUser, MUSMovie, String[], MUSMessage, MUSMessage)
  Constructor

Variables

m_user

```
public net.tabuleiro.nebulae.ServerUser m_user
```

m_mov

```
public net.tabuleiro.nebulae.MUSMovie m_mov
```

m_args

```
public java.lang.String m_args[]
```

m_msg

```
public net.tabuleiro.nebulae.MUSMessage m_msg
```

m_reply

```
public net.tabuleiro.nebulae.MUSMessage m_reply
```

Constructors

MUSDBMessage

```
 public MUSDBMessage(net.tabuleiro.nebulae.ServerUser user,
                     net.tabuleiro.nebulae.MUSMovie mov,
                     java.lang.String args[],
                     net.tabuleiro.nebulae.MUSMessage msg,
                     net.tabuleiro.nebulae.MUSMessage reply)
```

Constructor

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSDispatcher

```
net.tabuleiro.nebulae.MUSDispatcher
```

public class **MUSDispatcher**
extends java.lang.Object

This class is reserved for internal use of the Nebulae MultiUser Server

Constructor Index

- **MUSDispatcher**()

Method Index

- **handleMsg**(ServerUser, MUSMessage)
  Method not exposed to server side scripting

Constructors

MUSDispatcher

```
 public MUSDispatcher()
```

Methods

handleMsg

```
 public void handleMsg(net.tabuleiro.nebulae.ServerUser user,
                       net.tabuleiro.nebulae.MUSMessage msg)
```

Method not exposed to server side scripting

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSErrorCode

```
net.tabuleiro.nebulae.MUSErrorCode
```

public class **MUSErrorCode**
extends java.lang.Exception

Class representing a MultiUser Server error code.

Variable Index

- **BadConnectionID**
- **BadParameter**
- **ConnectionDuplicate**
- **ConnectionRefused**
- **DatabaseAddUser**
- **DatabaseDataNotFound**
- **DatabaseDataRecordNotUnique**
- **DatabaseError**
- **DatabaseLocked**
- **DatabaseMovedPastLimits**
- **DatabaseNoConfigurationFile**
- **DatabaseNoCurrentDB**
- **DatabaseNoCurrentRecord**
- **DatabaseNoCurrentTag**
- **DatabaseRead**
- **DatabaseRecordNotExists**
- **DatabaseRecordNotLocked**
- **DatabaseUserIDNotFound**
- **DatabaseWrite**
- **DataConcurrencyError**
- **ErrorJoiningGroup**
- **ErrorLeavingGroup**
- **IncomingDataLost**
- **InvalidGroupName**
- **InvalidMessage**
- **InvalidMessageFormat**
- **InvalidMessageLength**
- **InvalidMessageRecipient**
- **InvalidMovieID**
- **InvalidNumberOfMessageRecipients**
- **InvalidPassword**
- **InvalidServerCommand**
- **InvalidServerInitFile**
- **InvalidServerName**

- **InvalidUserID**
- **m_errCode**
- **MessageContainsErrorInfo**
- **MessageMissing**
- **MessageTooLarge**
- **NoConnectionsAvailable**
- **NoCurrentConnection**
- **NoError**
- **NoSocketManager**
- **NotPermittedWithUserLevel**
- **NoWaitingMessage**
- **OperationNotAllowed**
- **RequestedDataNotFound**
- **ServerCloseFailed**
- **ServerInitializationFailed**
- **ServerInternalError**
- **ServerSendFailed**
- **UDPSocketError**
- **Unknown**
- **UnknownInternalError**
- **WrongNumberOfParams**

Constructor Index

- **MUSErrorCode**(int)

Variables

NoError

```
public static final int NoError
```

Unknown

```
public static final int Unknown
```

InvalidMovieID

```
public static final int InvalidMovieID
```

InvalidUserID

```
public static final int InvalidUserID
```

InvalidPassword

```
public static final int InvalidPassword
```

● **IncomingDataLost**

```
public static final int IncomingDataLost
```

● **InvalidServerName**

```
public static final int InvalidServerName
```

● **NoConnectionsAvailable**

```
public static final int NoConnectionsAvailable
```

● **BadParameter**

```
public static final int BadParameter
```

● **NoSocketManager**

```
public static final int NoSocketManager
```

● **NoCurrentConnection**

```
public static final int NoCurrentConnection
```

● **NoWaitingMessage**

```
public static final int NoWaitingMessage
```

● **BadConnectionID**

```
public static final int BadConnectionID
```

● **WrongNumberOfParams**

```
public static final int WrongNumberOfParams
```

● **UnknownInternalError**

```
public static final int UnknownInternalError
```

● **ConnectionRefused**

```
public static final int ConnectionRefused
```

● **MessageTooLarge**

```
public static final int MessageTooLarge
```

● **InvalidMessageFormat**

```
public static final int InvalidMessageFormat
```

● **InvalidMessageLength**

```
public static final int InvalidMessageLength
```

● **MessageMissing**

```
public static final int MessageMissing
```

● **ServerInitializationFailed**

```
public static final int ServerInitializationFailed
```

● **ServerSendFailed**

```
public static final int ServerSendFailed
```

● **ServerCloseFailed**

```
public static final int ServerCloseFailed
```

● **ConnectionDuplicate**

```
public static final int ConnectionDuplicate
```

● **InvalidNumberOfMessageRecipients**

```
public static final int InvalidNumberOfMessageRecipients
```

● **InvalidMessageRecipient**

```
public static final int InvalidMessageRecipient
```

● **InvalidMessage**

```
public static final int InvalidMessage
```

● **ServerInternalError**

```
public static final int ServerInternalError
```

● **ErrorJoiningGroup**

```
public static final int ErrorJoiningGroup
```

● **ErrorLeavingGroup**

```
public static final int ErrorLeavingGroup
```

● **InvalidGroupName**

```
public static final int InvalidGroupName
```

● **InvalidServerCommand**

```
public static final int InvalidServerCommand
```

NotPermittedWithUserLevel

```
public static final int NotPermittedWithUserLevel
```

DatabaseError

```
public static final int DatabaseError
```

InvalidServerInitFile

```
public static final int InvalidServerInitFile
```

DatabaseWrite

```
public static final int DatabaseWrite
```

DatabaseRead

```
public static final int DatabaseRead
```

DatabaseUserIDNotFound

```
public static final int DatabaseUserIDNotFound
```

DatabaseAddUser

```
public static final int DatabaseAddUser
```

DatabaseLocked

```
public static final int DatabaseLocked
```

DatabaseDataRecordNotUnique

```
public static final int DatabaseDataRecordNotUnique
```

DatabaseNoCurrentRecord

```
public static final int DatabaseNoCurrentRecord
```

DatabaseRecordNotExists

```
public static final int DatabaseRecordNotExists
```

DatabaseMovedPastLimits

```
public static final int DatabaseMovedPastLimits
```

DatabaseDataNotFound

```
public static final int DatabaseDataNotFound
```

DatabaseNoCurrentTag

```
public static final int DatabaseNoCurrentTag
```

DatabaseNoCurrentDB

```
public static final int DatabaseNoCurrentDB
```

DatabaseNoConfigurationFile

```
public static final int DatabaseNoConfigurationFile
```

DatabaseRecordNotLocked

```
public static final int DatabaseRecordNotLocked
```

OperationNotAllowed

```
public static final int OperationNotAllowed
```

RequestedDataNotFound

```
public static final int RequestedDataNotFound
```

MessageContainsErrorInfo

```
public static final int MessageContainsErrorInfo
```

DataConcurrencyError

```
public static final int DataConcurrencyError
```

UDPSocketError

```
public static final int UDPSocketError
```

m_errCode

```
public int m_errCode
```

Constructors

MUSErrorCode

```
public MUSErrorCode(int msg)
```

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSGroup

```
net.tabuleiro.nebulae.MUSGroup
```

public class **MUSGroup**
extends java.lang.Object

This class is reserved for internal use of the Nebulae MultiUser Server
Scripts should use the ServerGroup interface

Variable Index

- **m_name**
  Member variable not exposed to server side scripting

Constructor Index

- **MUSGroup**()

Variables

m_name

```
public java.lang.String m_name
```

Member variable not exposed to server side scripting

Constructors

MUSGroup

```
public MUSGroup()
```

Class net.tabuleiro.nebulae.MUSLog

```
net.tabuleiro.nebulae.MUSLog
```

public class **MUSLog**
extends java.lang.Object

Class to log messages to the server output (console or text file)

Variable Index

- **kDB**
  Type of log message : database
- **kDeb**
  Type of log message : debug
- **kDebWarn**
  Type of log message : debug warning
- **kGrp**
  Type of log message : group
- **kMov**
  Type of log message : movie
- **kMsgErr**
  Type of log message : error in message handling
- **kScr**
  Type of log message : scripting
- **kSrv**
  Type of log message : server
- **kSys**
  Type of log message : system
- **kUsr**
  Type of log message : user
- **m_LogLevel**
  Bitmaks value set automatically by Nebulae from the config file log directives

Constructor Index

- **MUSLog**()

Method Index

- **Log**(Exception, int)
  Logs a java exception to output.
- **Log**(String, int)

Log as message to output.

- **setLogLevel**(int)

  Integer value represent the log level bitmask

Variables

● **m_LogLevel**

```
public static int m_LogLevel
```

Bitmaks value set automatically by Nebulae from the config file log directives

● **kSys**

```
public static final int kSys
```

Type of log message : system

● **kSrv**

```
public static final int kSrv
```

Type of log message : server

● **kMov**

```
public static final int kMov
```

Type of log message : movie

● **kGrp**

```
public static final int kGrp
```

Type of log message : group

● **kUsr**

```
public static final int kUsr
```

Type of log message : user

● **kDB**

```
public static final int kDB
```

Type of log message : database

● **kMsgErr**

```
public static final int kMsgErr
```

Type of log message : error in message handling

kScr

```
public static final int kScr
```

Type of log message : scripting

kDeb

```
public static final int kDeb
```

Type of log message : debug

kDebWarn

```
public static final int kDebWarn
```

Type of log message : debug warning

Constructors

MUSLog

```
public MUSLog()
```

Methods

setLogLevel

```
public static void setLogLevel(int level)
```

Integer value represent the log level bitmask

Log

```
public static void Log(java.lang.String str,
                       int level)
```

Log as message to output. Integer parameter is the type of message, for example MUSLog.kSys.

Log

```
public static void Log(java.lang.Exception e,
                       int level)
```

Logs a java exception to output. Integer parameter is the type of message, for example MUSLog.kDB.

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSMessage

```
net.tabuleiro.nebulae.MUSMessage
```

public class **MUSMessage**
extends java.lang.Object

Class representing a message formatted according to the Shockwave MultiUser Server specs.
See technote 15465 "Shockwave Multiuser protocol description" at www.macromedia.com/support for more information about the internal structure of a Shockwave binary message. Shockwave is a trademark of Macromedia, Inc. All rights reserved.

Variable Index

- **m_errCode**
  Message error code, represented as a MUSErrorCode type (for example MUSErrorCode.NoError)
- **m_header**
  Default MUS message header, included automatically with each message.
- **m_msgContent**
  The content part of this message.
- **m_recptID**
  A MUSMsgHeaderStringList object containing one or more MUSMsgHeaderStrings, each corresponding to one intended recipient for this message.
- **m_senderID**
  A single MUSMsgHeaderString object corresponding to the name of the message's sender.
- **m_subject**
  A single MUSMsgHeaderString object corresponding to the message's subject.
- **m_timeStamp**
  Message timestamp.
- **m_udp**
  UDP flag for this message.

Constructor Index

- **MUSMessage**()
  Default Constructor
- **MUSMessage**(boolean, String)
  Default Constructor for login messages.
- **MUSMessage**(MUSMessage)
  Constructor.

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractMUSMessage**(byte[])
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **toDatagramPacket**(InetAddress, int)
  Reserved for internal use of the Nebulae MultiUser Server.

Variables

m_header

```
public static final byte m_header[]
```

Default MUS message header, included automatically with each message.

m_errCode

```
public int m_errCode
```

Message error code, represented as a MUSErrorCode type (for example MUSErrorCode.NoError)

m_timeStamp

```
public int m_timeStamp
```

Message timestamp. Usually set automatically by Nebulae's message dispatcher.

m_subject

```
public net.tabuleiro.nebulae.MUSMsgHeaderString m_subject
```

A single MUSMsgHeaderString object corresponding to the message's subject.

m_senderID

```
public net.tabuleiro.nebulae.MUSMsgHeaderString m_senderID
```

A single MUSMsgHeaderString object corresponding to the name of the message's sender.

m_recptID

```
public net.tabuleiro.nebulae.MUSMsgHeaderStringList m_recptID
```

A MUSMsgHeaderStringList object containing one or more MUSMsgHeaderStrings, each corresponding to one intended recipient for this message.

m_msgContent

```
public net.tabuleiro.nebulae.LValue m_msgContent
```

The content part of this message. Content is always one single LValue, but it may be a linear or property list including other LValues.

m_udp

```
public boolean m_udp
```

UDP flag for this message. When set to TRUE the dispatcher will attempt to deliver the message using the UDP connection channel.

Constructors

MUSMessage

```
public MUSMessage()
```

Default Constructor

MUSMessage

```
public MUSMessage(net.tabuleiro.nebulae.MUSMessage msg)
```

Constructor. Clones another message.

MUSMessage

```
public MUSMessage(boolean islogin,
                  java.lang.String logininfo)
```

Default Constructor for login messages. Reserved for internal use of the Nebulae MultiUser Server.

Methods

extractMUSMessage

```
public void extractMUSMessage(byte rawmsg[])
```

Reserved for internal use of the Nebulae MultiUser Server.

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

● getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

● toDatagramPacket

```
public java.net.DatagramPacket toDatagramPacket(java.net.InetAddress addr,
                                                int port)
```

Reserved for internal use of the Nebulae MultiUser Server.

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSMovie

```
net.tabuleiro.nebulae.MUSMovie
```

public class **MUSMovie**
extends java.lang.Object

This class is reserved for internal use of the Nebulae MultiUser Server
Scripts should use the ServerMovie interface

Variable Index

- **m_dispatcher**
  Member variable not exposed to server side scripting

Constructor Index

- **MUSMovie**()

Variables

m_dispatcher

```
public net.tabuleiro.nebulae.MUSDispatcher m_dispatcher
```

Member variable not exposed to server side scripting

Constructors

MUSMovie

```
public MUSMovie()
```

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSMsgHeaderString

`net.tabuleiro.nebulae.MUSMsgHeaderString`

public class **MUSMsgHeaderString**
extends java.lang.Object

Class representing a message header String. Direct methods for conversion to/from a Java string are provided.

Constructor Index

- **MUSMsgHeaderString**()
  Default Constructor
- **MUSMsgHeaderString**(String)
  Constructs a MUSMsgHeaderString from a Java String.

Method Index

- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **extractMUSMsgHeaderString**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **toString**()
  Returns this MUSMsgHeaderString as a Java String.

Constructors

MUSMsgHeaderString

```
 public MUSMsgHeaderString(java.lang.String initString)
```

Constructs a MUSMsgHeaderString from a Java String.

MUSMsgHeaderString

```
 public MUSMsgHeaderString()
```

Default Constructor

Methods

extractMUSMsgHeaderString

```
public int extractMUSMsgHeaderString(byte rawBytes[],
                                     int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

toString

```
public java.lang.String toString()
```

Returns this MUSMsgHeaderString as a Java String.

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSMsgHeaderStringList

```
net.tabuleiro.nebulae.MUSMsgHeaderStringList
```

public class **MUSMsgHeaderStringList**
extends java.lang.Object

Class representing a list of MUSMsgHeaderString objects, stored as a Java Vector.

Variable Index

- **m_stringlist**
  Public vector element storing the MUSMsgHeaderString members.

Constructor Index

- **MUSMsgHeaderStringList**()
  Default Constructor

Method Index

- **addElement**(MUSMsgHeaderString)
  Adds a MUSMsgHeaderString to the list
- **dump**()
  Reserved for internal use of the Nebulae MultiUser Server.
- **elements**()
  Retrieves an Enumeration object containing the MUSMsgHeaderString elements.
- **extractMUSMsgHeaderStringList**(byte[], int)
  Reserved for internal use of the Nebulae MultiUser Server.
- **getBytes**()
  Reserved for internal use of the Nebulae MultiUser Server.

Variables

m_stringlist

```
 public java.util.Vector m_stringlist
```

Public vector element storing the MUSMsgHeaderString members. It is safe to access the elements directly.

Constructors

MUSMsgHeaderStringList

```
public MUSMsgHeaderStringList()
```

Default Constructor

Methods

addElement

```
public boolean addElement(net.tabuleiro.nebulae.MUSMsgHeaderString elem)
```

Adds a MUSMsgHeaderString to the list

Parameters:

elem - MUSMsgHeaderString to add

Returns:

boolean

elements

```
public java.util.Enumeration elements()
```

Retrieves an Enumeration object containing the MUSMsgHeaderString elements.

extractMUSMsgHeaderStringList

```
public int extractMUSMsgHeaderStringList(byte rawBytes[],
                                         int offset)
```

Reserved for internal use of the Nebulae MultiUser Server.

dump

```
public void dump()
```

Reserved for internal use of the Nebulae MultiUser Server.

getBytes

```
public byte[] getBytes()
```

Reserved for internal use of the Nebulae MultiUser Server.

All Packages   Class Hierarchy   This Package   Previous   Next   Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.MUSServer

```
net.tabuleiro.nebulae.MUSServer
```

public class **MUSServer**
extends java.lang.Object

This class is reserved for internal use of the Nebulae MultiUser Server
Scripts should use the ServerObject interface

Constructor Index

- **MUSServer**()

Constructors

● **MUSServer**

```
 public MUSServer()
```

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.Nebulae

```
net.tabuleiro.nebulae.Nebulae
```

public class **Nebulae**
extends java.lang.Object

This main Nebulae class

Constructor Index

● **Nebulae**()

Method Index

● **main**(String[])
 Entry point for the application
● **restart**()
 Reserved for internal use of the Nebulae MultiUser Server

Constructors

● Nebulae

```
public Nebulae()
```

Methods

● main

```
public static void main(java.lang.String args[])
```

Entry point for the application

● restart

```
public static void restart()
```

Reserved for internal use of the Nebulae MultiUser Server

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.PlayerNotFoundException

```
net.tabuleiro.nebulae.PlayerNotFoundException
```

public class **PlayerNotFoundException**
extends java.lang.Exception
implements java.io.Serializable

Constructor Index

- **PlayerNotFoundException**(String)

Constructors

PlayerNotFoundException

```
 public PlayerNotFoundException(java.lang.String msg)
```

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.PropertyNotFoundException

```
net.tabuleiro.nebulae.PropertyNotFoundException
```

public class **PropertyNotFoundException**
extends java.lang.Exception
implements java.io.Serializable

Constructor Index

- **PropertyNotFoundException**(String)

Constructors

PropertyNotFoundException

```
 public PropertyNotFoundException(java.lang.String msg)
```

All Packages Class Hierarchy This Package Previous Next Index

Interface net.tabuleiro.nebulae.ServerGroup

public abstract interface **ServerGroup**

Interface representing a group on a movie.
Scripts can use methods of the ServerMovie interface to retrieve pointers to existing groups.

Method Index

- **addUser**(ServerUser)
  Adds an user to this server group
- **getServerUser**(int)
  Retrieves a pointer to a ServerUser object representing a user connected to the movie.
- **getServerUser**(String)
  Retrieves a pointer to a ServerUser object representing a user connected to the movie.
- **name**()
  Returns the name of the group as a String
- **persists**()
  Checks if the group is set to persist on the movie even when no users members of it.
- **removeUser**(ServerUser)
  Removes an user from this server group.
- **sendMessage**(MUSMessage)
  Sends a message to all users that are members of this group
- **serverUserCount**()
  Gets the number of users in this group.
- **setpersists**(boolean)
  Toggles the group persistent flag for this group.
- **setuserLimit**(int)
  Sets the maximum number of users allowed to join this group.
- **userLimit**()
  Returns the maximum number of users allowed to join this group.

Methods

getServerUser

```
 public abstract net.tabuleiro.nebulae.ServerUser getServerUser(java.lang.String username) thro
```

Retrieves a pointer to a ServerUser object representing a user connected to the movie.

Parameters:

username - String with the name of the user to be returned

Returns:

ServerUser pointer, or throws a UserNotFoundException

getServerUser

```
public abstract net.tabuleiro.nebulae.ServerUser getServerUser(int useridx) throws net.tabulei
```

Retrieves a pointer to a ServerUser object representing a user connected to the movie.

Parameters:

useridx - index of the user to be returned

Returns:

ServerUser pointer, or throws a UserNotFoundException

addUser

```
public abstract void addUser(net.tabuleiro.nebulae.ServerUser oneuser) throws net.tabuleiro.ne
```

Adds an user to this server group

Parameters:

oneuser - ServerUser object to be added

removeUser

```
public abstract void removeUser(net.tabuleiro.nebulae.ServerUser oneuser)
```

Removes an user from this server group.

Parameters:

oneuser - ServerUser object to be removed

serverUserCount

```
public abstract int serverUserCount()
```

Gets the number of users in this group.

Returns:

the number of users

sendMessage

```
public abstract void sendMessage(net.tabuleiro.nebulae.MUSMessage msg)
```

Sends a message to all users that are members of this group

Parameters:

msg - Message to send

name

```
public abstract java.lang.String name()
```

Returns the name of the group as a String

userLimit

```
public abstract int userLimit()
```

Returns the maximum number of users allowed to join this group.
By default there is no limit on the number of users, unless the GroupSizeLimits directive in specified in a movie configuration file.
Returns:
the number of users allowed to join the group, or -1 for unlimited users

setuserLimit

```
public abstract void setuserLimit(int level)
```

Sets the maximum number of users allowed to join this group.
By default there is no limit on the number of users, unless the GroupSizeLimits directive in specified in a movie configuration file.
Parameters:
level - the number of users allowed to join the group, or -1 for unlimited users

persists

```
public abstract boolean persists()
```

Checks if the group is set to persist on the movie even when no users members of it. Groups are not persistent by default.
Returns:
TRUE if the group is set to persist, FALSE otherwise

setpersists

```
public abstract void setpersists(boolean persistflag)
```

Toggles the group persistent flag for this group. Persistent groups are not destroyed even when no users are connected to it.
Parameters:
persistflag - TRUE if the group needs to persist on the movie, FALSE otherwise

---

All Packages Class Hierarchy This Package Previous Next Index

Interface net.tabuleiro.nebulae.ServerMovie

public abstract interface **ServerMovie**

Interface representing a movie connected to the Nebulae server.
ServerSideScript classes can use the serverMovie() method to retrieve a pointer to the ServerMovie object that created them.
Scripts can also use methods of the ServerObject interface to retrieve pointers to other movies connected to the server.

Method Index

- **createServerGroup**(String)
  Creates a new group on the movie and returns a pointer to it.
- **deleteServerGroup**(String)
  Deletes a group from the movie.
- **disableGroup**(String)
  Disables this group so new users can not join it.
- **enableGroup**(String)
  Enables this group to receive new users.
- **getServerGroup**(int)
  Retrieves a pointer to a ServerGroup object representing a group that exists on this ServerMovie.
- **getServerGroup**(String)
  Retrieves a pointer to a ServerGroup object representing a group that exists on this ServerMovie.
- **name**()
  Returns the name of the movie as a String
- **persists**()
  Checks if the movie is set to persist on the server even when no users are connected to it.
- **serverGroupCount**()
  Gets the number of groups in this movie.
- **serverUserCount**()
  Gets the number of users connected to this movie.
- **setpersists**(boolean)
  Toggles the movie persistent flag for this movie.
- **setuserLevel**(int)
  Sets the default user levels for users that connect to this movie and do not have an user account in the DB.
- **userLevel**()
  Gets the default user levels for users that connect to this movie and do not have an user account in the DB.

Methods

- **getServerGroup**

```
public abstract net.tabuleiro.nebulae.ServerGroup getServerGroup(java.lang.String groupname) t
```

Retrieves a pointer to a ServerGroup object representing a group that exists on this ServerMovie.

Parameters:

groupname - String with the name of the group to be returned

Returns:

ServerGroup pointer, or throws a GroupNotFoundException

getServerGroup

```
public abstract net.tabuleiro.nebulae.ServerGroup getServerGroup(int groupidx) throws net.tabu
```

Retrieves a pointer to a ServerGroup object representing a group that exists on this ServerMovie.

Parameters:

groupidx - index of the movie to be returned

Returns:

ServerGroup pointer, or throws a GroupNotFoundException

createServerGroup

```
public abstract net.tabuleiro.nebulae.ServerGroup createServerGroup(java.lang.String groupname
```

Creates a new group on the movie and returns a pointer to it.

Parameters:

groupname - String with the name of the group to be created

Returns:

ServerGroup pointer, or throws a MUSErrorCode if the creation fails

deleteServerGroup

```
public abstract void deleteServerGroup(java.lang.String groupname)
```

Deletes a group from the movie.

Parameters:

groupname - String with the name of the group to be deleted

serverGroupCount

```
public abstract int serverGroupCount()
```

Gets the number of groups in this movie.

Returns:

the number of groups

serverUserCount

```
public abstract int serverUserCount()
```

Gets the number of users connected to this movie.

Returns:

the number of users

● name

```
public abstract java.lang.String name()
```

Returns the name of the movie as a String

● userLevel

```
public abstract int userLevel()
```

Gets the default user levels for users that connect to this movie and do not have an user account in the DB.
This value can be configured by the DefaultUserLevel directive in a movie configuration file.

● setuserLevel

```
public abstract void setuserLevel(int level)
```

Sets the default user levels for users that connect to this movie and do not have an user account in the DB.
When this method is used it overrides the value configured by the DefaultUserLevel directive in a movie configuration file.

● enableGroup

```
public abstract void enableGroup(java.lang.String gname)
```

Enables this group to receive new users. Groups are enabled by default.

● disableGroup

```
public abstract void disableGroup(java.lang.String gname)
```

Disables this group so new users can not join it.

● persists

```
public abstract boolean persists()
```

Checks if the movie is set to persist on the server even when no users are connected to it. Movies are not persistent by default.
Returns:
TRUE if the movie is set to persist, FALSE otherwise

● setpersists

```
public abstract void setpersists(boolean persistflag)
```

Toggles the movie persistent flag for this movie. Persistent movies are not destroyed even when no users are connected to it.
Parameters:
persistflag - TRUE if the movie needs to persist on the server, FALSE otherwise

All Packages Class Hierarchy This Package Previous Next Index

Interface net.tabuleiro.nebulae.ServerObject

public abstract interface **ServerObject**

Interface representing the Nebulae server instance.
ServerSideScript classes can use the serverObject() method to retrieve a pointer to a ServerObject.

Method Index

- **createServerMovie**(String)
  Creates a new movie on the server and returns a pointer to it.
- **deleteServerMovie**(String)
  Deletes a movie from the server.
- **getServerMovie**(int)
  Retrieves a pointer to a ServerMovie object active on the server.
- **getServerMovie**(String)
  Retrieves a pointer to a ServerMovie object active on the server.
- **getServerUserDatabase**()
  Returns a ponter to the ServerUserDatabase interface representing the default user authentication table.
- **getSQLGateway**()
  Returns a ponter to the SQLGateway interface representing the default SQL connection.
- **language**()
  This method is reserved.
- **path**()
  Gets the path of the server executable in the system
- **put**(String)
  Displays a message to the server active output (log file or terminal window)
- **serverMovieCount**()
  Gets the number of active movies in the server.
- **setuserLevel**(int)
  Sets the default user levels for users that connect to the server and do not have an user account in the DB.
- **timeStamp**()
  Gets the current server time as an int
- **timeString**()
  Gets the current server time formatted as a String
- **userLevel**()
  Gets the default user levels for users that connect to the server and do not have an user account in the DB.

Methods

- **put**

```
public abstract void put(java.lang.String msg)
```

Displays a message to the server active output (log file or terminal window)

getServerMovie

```
public abstract net.tabuleiro.nebulae.ServerMovie getServerMovie(java.lang.String moviename) t
```

Retrieves a pointer to a ServerMovie object active on the server.

Parameters:

moviename - String with the name of the movie to be returned

Returns:

ServerMovie pointer, or throws a MovieNotFoundException

getServerMovie

```
public abstract net.tabuleiro.nebulae.ServerMovie getServerMovie(int movieidx) throws net.tabu
```

Retrieves a pointer to a ServerMovie object active on the server.

Parameters:

movieidx - index of the movie to be returned

Returns:

ServerMovie pointer, or throws a MovieNotFoundException

createServerMovie

```
public abstract net.tabuleiro.nebulae.ServerMovie createServerMovie(java.lang.String moviename
```

Creates a new movie on the server and returns a pointer to it.

Parameters:

moviename - String with the name of the movie to be created

Returns:

ServerMovie pointer, or throws a MUSErrorCode if the creation fails

deleteServerMovie

```
public abstract void deleteServerMovie(java.lang.String moviename)
```

Deletes a movie from the server.

Parameters:

moviename - String with the name of the movie to be deleted

serverMovieCount

```
public abstract int serverMovieCount()
```

Gets the number of active movies in the server.

Returns:

the number of movies

path

```
public abstract java.lang.String path()
```

Gets the path of the server executable in the system

timeString

```
public abstract java.lang.String timeString()
```

Gets the current server time formatted as a String

timeStamp

```
public abstract int timeStamp()
```

Gets the current server time as an int

language

```
public abstract int language()
```

This method is reserved. Current implementation returns 0 for the English version of Nebulae.

userLevel

```
public abstract int userLevel()
```

Gets the default user levels for users that connect to the server and do not have an user account in the DB.
This value can be configured by the DefaultUserLevel directive in Nebulae.cfg.

setuserLevel

```
public abstract void setuserLevel(int level)
```

Sets the default user levels for users that connect to the server and do not have an user account in the DB.
When this method is used it overrides the value configured by the DefaultUserLevel directive in Nebulae.cfg.

getSQLGateway

```
public abstract net.tabuleiro.nebulae.SQLGateway getSQLGateway()
```

Returns a ponter to the SQLGateway interface representing the default SQL connection.
Please consult the documentation of SQLGateway for more information.

getServerUserDatabase

```
public abstract net.tabuleiro.nebulae.ServerUserDatabase getServerUserDatabase()
```

Returns a ponter to the ServerUserDatabase interface representing the default user authentication table.

Please consult the documentation of ServerUserDatabase for more information.

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.ServerSideScript

```
net.tabuleiro.nebulae.ServerSideScript
```

public class **ServerSideScript**
extends java.lang.Object
implements net.tabuleiro.nebulae.ServerUser

Base class representing a ServerSideScript object. All server side scripts must extend this class.
The ServerSideScript class implements the ServerUser interface, so scripts can join movies and send/receive messages as if they were a connected user.
Server side scripts are instantiated when a movie is created on the server. Classes are mapped to movies according to the Scriptmap.cfg file.
Messages that are addressed to a server side script should be sent with the system.script.* prefix.

Variable Index

- **m_name**
  The name of the script object.
- **m_userlevel**
  The user access level for this script.

Constructor Index

- **ServerSideScript**()
  Constructor.

Method Index

- **creationTime**()
  Gets this scripts's creationTime on the server.
- **deleteUser**()
  Scripts should not call this method, it is reserved for internal use of the Nebulae MultiUser Server.
- **getGroupNames**()
  Gets a list of the groups this script is a member of.
- **getGroupsCount**()
  Gets the number of groups this script is a member of.
- **groupCreate**(ServerGroup)
  Called by the Nebulae server when a group is created in the movie associated with this server side script.
- **groupDelete**(ServerGroup)
  Called by the Nebulae server when a group is deleted from the movie associated with this server side script.

- **groupJoin**(ServerUser, ServerGroup)
  Called by the Nebulae server when an user joins a group in the movie associated with this server side script.
- **groupJoined**(ServerGroup)
  Called by the Nebulae server when the script joins a group.
- **groupLeave**(ServerUser, ServerGroup)
  Called by the Nebulae server when an user leaves a group in the movie associated with this server side script.
- **groupLeft**(ServerGroup)
  Called by the Nebulae server when the script leaves a group.
- **incomingMessage**(ServerUser, MUSMessage)
  Called by the Nebulae server when a message addressed to this script object arrives.
- **initScript**(ServerObject, ServerMovie)
  Reserved for internal use of the Nebulae MultiUser Server.
- **ipAddress**()
  Returns "localhost" for server side scripts.
- **name**()
  Gets the script name.
- **postMessage**(MUSMessage)
  Posts a message to the Nebulae server dispatcher.
- **scriptCreate**()
  Called by the Nebulae server when the script object is created on the server.
- **scriptDelete**()
  Called by the Nebulae server when the script object is destroyed on the server.
- **sendMessage**(MUSMessage)
  Sends a message to this script directly.
- **serverMovie**()
  Retrieves a pointer to the server movie object that created this script.
- **serverObject**()
  Retrieves a pointer to a ServerObject instance representing the current server.
- **setuserLevel**(int)
  Sets the user access level for this script.
- **userLevel**()
  Gets the user access level for this script.
- **userLogOff**(ServerUser)
  Called by the Nebulae server when an user is disconnected from the movie associated with this server side script.
- **userLogOn**(ServerUser)
  Called by the Nebulae server when an user connects to the movie associated with this server side script.

Variables

● **m_name**

```
public java.lang.String m_name
```

The name of the script object.

m_userlevel

```
public int m_userlevel
```

The user access level for this script. Default is 100.

Constructors

ServerSideScript

```
public ServerSideScript()
```

Constructor. Reserved for internal use of the Nebulae MultiUser Server.

Methods

serverObject

```
public net.tabuleiro.nebulae.ServerObject serverObject()
```

Retrieves a pointer to a ServerObject instance representing the current server.

initScript

```
public void initScript(net.tabuleiro.nebulae.ServerObject srv,
                       net.tabuleiro.nebulae.ServerMovie mov)
```

Reserved for internal use of the Nebulae MultiUser Server.
Scripts should implement the scriptCreate() method to perform initialization tasks.

incomingMessage

```
public void incomingMessage(net.tabuleiro.nebulae.ServerUser user,
                            net.tabuleiro.nebulae.MUSMessage msg)
```

Called by the Nebulae server when a message addressed to this script object arrives.
Scripts should implement this method to receive message from other users. Messages should be addressed to system.script.*, and are passed intact for processing.

Parameters:

user - ServerUser reference, representing the user that has sent the message.
msg - Message to be processed.

scriptCreate

```
public void scriptCreate()
```

Called by the Nebulae server when the script object is created on the server. Scripts should implement this method to perform initialization tasks.

scriptDelete

```
public void scriptDelete()
```

Called by the Nebulae server when the script object is destroyed on the server. Scripts should implement this method to perform cleanup tasks.

userLogOn

```
public void userLogOn(net.tabuleiro.nebulae.ServerUser usr)
```

Called by the Nebulae server when an user connects to the movie associated with this server side script.

Parameters:

usr - ServerUser reference

userLogOff

```
public void userLogOff(net.tabuleiro.nebulae.ServerUser usr)
```

Called by the Nebulae server when an user is disconnected from the movie associated with this server side script.

Parameters:

usr - ServerUser reference

groupCreate

```
public void groupCreate(net.tabuleiro.nebulae.ServerGroup grp)
```

Called by the Nebulae server when a group is created in the movie associated with this server side script.

Parameters:

grp - ServerGroup reference

groupDelete

```
public void groupDelete(net.tabuleiro.nebulae.ServerGroup grp)
```

Called by the Nebulae server when a group is deleted from the movie associated with this server side script.

Parameters:

grp - ServerGroup reference

groupJoin

```
public void groupJoin(net.tabuleiro.nebulae.ServerUser usr,
                      net.tabuleiro.nebulae.ServerGroup grp)
```

Called by the Nebulae server when an user joins a group in the movie associated with this server side script.

Parameters:

usr - ServerUser reference

grp - ServerGroup reference

groupLeave

```
public void groupLeave(net.tabuleiro.nebulae.ServerUser usr,
                       net.tabuleiro.nebulae.ServerGroup grp)
```

Called by the Nebulae server when an user leaves a group in the movie associated with this server side script.

Parameters:

usr - ServerUser reference
grp - ServerGroup reference

sendMessage

```
public void sendMessage(net.tabuleiro.nebulae.MUSMessage msg)
```

Sends a message to this script directly. Message will be received by the incomingMessage() method.
This method is implemented for compatibility with the ServerUser interface.

Parameters:

msg - Message to send

postMessage

```
public void postMessage(net.tabuleiro.nebulae.MUSMessage msg)
```

Posts a message to the Nebulae server dispatcher.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

Parameters:

msg - Message to post

name

```
public java.lang.String name()
```

Gets the script name.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

userLevel

```
public int userLevel()
```

Gets the user access level for this script.
By default all scripts are instantiated with full privileges (userlevel 100).
This method is part of the ServerUser interface implemented by ServerSideScript objects.

setuserLevel

```
public void setuserLevel(int level)
```

Sets the user access level for this script.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

Parameters:

level - new user access level

serverMovie

```
public net.tabuleiro.nebulae.ServerMovie serverMovie()
```

Retrieves a pointer to the server movie object that created this script.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

Returns:

ServerMovie pointer

creationTime

```
public long creationTime()
```

Gets this scripts's creationTime on the server.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

ipAddress

```
public java.lang.String ipAddress()
```

Returns "localhost" for server side scripts.
This method is implemented for compatibility with the ServerUser interface.

getGroupNames

```
public java.util.Vector getGroupNames()
```

Gets a list of the groups this script is a member of.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

Returns:

Java Vector with the group names as Strings.

getGroupsCount

```
public int getGroupsCount()
```

Gets the number of groups this script is a member of.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

Returns:

the number of groups

deleteUser

```
public void deleteUser()
```

Scripts should not call this method, it is reserved for internal use of the Nebulae MultiUser Server.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

groupJoined

```
public void groupJoined(net.tabuleiro.nebulae.ServerGroup grp)
```

Called by the Nebulae server when the script joins a group.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

Parameters:

grp - ServerGroup reference

● groupLeft

```
public void groupLeft(net.tabuleiro.nebulae.ServerGroup grp)
```

Called by the Nebulae server when the script leaves a group.
This method is part of the ServerUser interface implemented by ServerSideScript objects.

Parameters:

grp - ServerGroup reference

All Packages Class Hierarchy This Package Previous Next Index

Interface net.tabuleiro.nebulae.ServerUser

public abstract interface **ServerUser**

Method Index

- **creationTime**()
  Gets this user's creationTime on the server, equivalent to the user login
- **deleteUser**()
  Deletes this user, disconnecting him from the server.
- **getGroupNames**()
  Gets a list of the groups this user is a member of.
- **getGroupsCount**()
  Gets the number of groups this user is a member of.
- **groupJoined**(ServerGroup)
  Called by the Nebulae server when the user joins a group.
- **groupLeft**(ServerGroup)
  Called by the Nebulae server when the user leaves a group.
- **ipAddress**()
  Gets this user's IP address as a String
- **name**()
  Returns the name of the user as a String
- **postMessage**(MUSMessage)
  Posts a message to the Nebulae server dispatcher.
- **sendMessage**(MUSMessage)
  Sends a message to this server user directly.
- **serverMovie**()
  Retrieves the ServerMovie object representing the movie this user is connected to.
- **setuserLevel**(int)
  Sets the user access level for this user.
- **userLevel**()
  Gets the user access level for this user

Methods

sendMessage

```
 public abstract void sendMessage(net.tabuleiro.nebulae.MUSMessage msg)
```

Sends a message to this server user directly.

Parameters:

msg - Message to send

postMessage

```
public abstract void postMessage(net.tabuleiro.nebulae.MUSMessage msg)
```

Posts a message to the Nebulae server dispatcher.
Scripts can use this method to post messages to the server as if they were a connected user, since the ServerSideScript class implements the ServerUser interface.

Parameters:

msg - Message to post

● name

```
public abstract java.lang.String name()
```

Returns the name of the user as a String

● userLevel

```
public abstract int userLevel()
```

Gets the user access level for this user

● setuserLevel

```
public abstract void setuserLevel(int level)
```

Sets the user access level for this user.

Parameters:

level - new user access level

● serverMovie

```
public abstract net.tabuleiro.nebulae.ServerMovie serverMovie()
```

Retrieves the ServerMovie object representing the movie this user is connected to.
Server side scripts can use this method to retrieve a pointer to the movie that created them.

● creationTime

```
public abstract long creationTime()
```

Gets this user's creationTime on the server, equivalent to the user login

● ipAddress

```
public abstract java.lang.String ipAddress()
```

Gets this user's IP address as a String

● getGroupNames

```
public abstract java.util.Vector getGroupNames()
```

Gets a list of the groups this user is a member of. By default all users are at least members of one group, @AllUsers

Returns:

Java Vector with the group names as Strings.

getGroupsCount

```
public abstract int getGroupsCount()
```

Gets the number of groups this user is a member of. By default all users are at least members of one group, @AllUsers

Returns:

the number of groups

deleteUser

```
public abstract void deleteUser()
```

Deletes this user, disconnecting him from the server.

groupJoined

```
public abstract void groupJoined(net.tabuleiro.nebulae.ServerGroup grp)
```

Called by the Nebulae server when the user joins a group.

Parameters:

grp - ServerGroup reference

groupLeft

```
public abstract void groupLeft(net.tabuleiro.nebulae.ServerGroup grp)
```

Called by the Nebulae server when the user leaves a group.

Parameters:

grp - ServerGroup reference

All Packages Class Hierarchy This Package Previous Next Index

Interface net.tabuleiro.nebulae.ServerUserDatabase

public abstract interface **ServerUserDatabase**

Interface used by Nebulae for user authentication, using the internal database system or an SQL source. The database backend used for user authentication is configured in the Nebulae.cfg file. Use the getServerUserDatabase() method of the ServerObject interface to acquire this object.

Method Index

- **createUser**(String, String, String)
  Creates an user record in the user database
- **deleteDBUser**(int)
  Deletes the user with the specified user id from the database.
- **getDBUser**(String)
  Returns the integer id for the user in the database.
- **getDBUserLevel**(int)
  Retrieves the user access level
- **getDBUserPassword**(int)
  Retrieves the user password
- **isEnabled**()
  Checks if the user database is enabled.
- **updateUserLastLoginTime**(int)
  Updates the last login time for this user in the database to the current time.

Methods

createUser

```
public abstract boolean createUser(java.lang.String usernamein,
                                   java.lang.String password,
                                   java.lang.String userlevel)
```

Creates an user record in the user database

Parameters:

usernamein - Username string, will be converted to uppercase for storage

password - Password string

userlevel - User access level as a string. Usual values are between 20 and 100. If the string specified can not be converted the default user level will be set.

Returns:

true if the user account is created successfully, false if an error occurs

updateUserLastLoginTime

```
public abstract void updateUserLastLoginTime(int userid)
```

Updates the last login time for this user in the database to the current time.
Parameters:
userid - User id integer retrieved by the getDBUser method.

getDBUser

```
public abstract int getDBUser(java.lang.String usernamein) throws net.tabuleiro.nebulae.DBExce
```

Returns the integer id for the user in the database.
Other methods use the user id for speed.
Parameters:
usernamein - Username string

deleteDBUser

```
public abstract boolean deleteDBUser(int userid)
```

Deletes the user with the specified user id from the database.
Parameters:
userid - User id integer retrieved by the getDBUser method.
Returns:
true if the user account is deleted successfully, false if an error occurs

getDBUserLevel

```
public abstract int getDBUserLevel(int userid) throws net.tabuleiro.nebulae.DBException
```

Retrieves the user access level
Parameters:
userid - User id integer retrieved by the getDBUser method.

getDBUserPassword

```
public abstract java.lang.String getDBUserPassword(int userid) throws net.tabuleiro.nebulae.DB
```

Retrieves the user password
Parameters:
userid - User id integer retrieved by the getDBUser method.

isEnabled

```
public abstract boolean isEnabled()
```

Checks if the user database is enabled.
Databases are enabled by default, but can be disabled by using the Nebulae.cfg file.

All Packages Class Hierarchy This Package Previous Next Index

Interface net.tabuleiro.nebulae.SQLGateway

public abstract interface **SQLGateway**

Interface used by Nebulae for communication with the default SQL database. This object is optimized to work with lists of LValues as parameters and return values to SQL queries. Scripting users will probably prefer to work with JDBC objects directly if data manipulation is required.
SQLGateway is only present if the EnableSQLDatabase is used in the Nebulae.cfg file. Use the getSQLGateway() method of the ServerObject interface to acquire this object.

Method Index

- **connect**(String, String, String, String)
  Connects to the default SQL database.
- **disconnect**()
  Disconnects from the SQL database Usually the connection is kept open for the entire duration of the Nebulae session
- **executeQuery**(String, LList)
  Executes an SQL query call
  Use question marks in the query as placeholders for values contained in the params list.
- **executeUpdate**(String, LList)
  Executes an SQL update call
  Use question marks in the query as placeholders for values contained in the params list.
- **getConnection**()
  Returns the default SQL database connection as a java.sql.Connection object.
- **isConnected**()
  Checks if the SQL database connection is alive

Methods

isConnected

```
public abstract boolean isConnected()
```

Checks if the SQL database connection is alive

getConnection

```
public abstract java.sql.Connection getConnection()
```

Returns the default SQL database connection as a java.sql.Connection object.
This is useful if the script author wants to call JDBC methods directly.

connect

```
public abstract boolean connect(java.lang.String sqldriver,
                                java.lang.String sqlurl,
                                java.lang.String sqluser,
                                java.lang.String sqlpassword)
```

Connects to the default SQL database.
This method is usually not necessary since Nebulae establishes the connection automatically using the following directives from Nebulae.cfg:
SQLDatabaseDriver
SQLDatabaseURL
SQLDatabaseUsername
SQLDatabasePassword

Parameters:

sqldriver - JDBC database driver
sqlurl - JDBC database URL
sqluser - Username for connection
sqlpassword - Password for connection

disconnect

```
public abstract void disconnect()
```

Disconnects from the SQL database Usually the connection is kept open for the entire duration of the Nebulae session

executeUpdate

```
public abstract boolean executeUpdate(java.lang.String sqlquery,
                                      net.tabuleiro.nebulae.LList params)
```

Executes an SQL update call
Use question marks in the query as placeholders for values contained in the params list.
Nebulae automatically maps the LList values to the appropriate SQL type.
Example:
LList params= new LList();
params.addElement(new LString("John"));
params.addElement(new LInteger(25));
sqlgateway.executeUpdate("UPDATE USERS SET NAME=? WHERE USERID=?",params);

Parameters:

sqlquery - Prepared SQL query string.
LList - of parameters to the prepared statement.

Returns:

true if no SQL error occurs

executeQuery

```
public abstract net.tabuleiro.nebulae.LValue executeQuery(java.lang.String sqlquery,
                                                          net.tabuleiro.nebulae.LList params)
```

Executes an SQL query call
Use question marks in the query as placeholders for values contained in the params list.
Nebulae automatically maps the LList values to the appropriate SQL type.

Example:
```
LList params= new LList();
params.addElement(new LString("John"));
params.addElement(new LInteger(25));
LValue result = sqlgateway.executeQuery("SELECT LASTNAME, AGE FROM USERS WHERE FIRSTNAME=? AND AGE>?",params);
```

Parameters:

sqlquery - Prepared SQL query string.
LList - of parameters to the prepared statement.

Returns:

LList containing the query results as LValues.
Each row in the result is returned as an LList inside the main return list.
Sample output (Lingo formatted): [["Perkins",10],["Garcia",23]]

All Packages Class Hierarchy This Package Previous Next Index

All Packages Class Hierarchy This Package Previous Next Index

Class net.tabuleiro.nebulae.UserNotFoundException

```
net.tabuleiro.nebulae.UserNotFoundException
```

public class **UserNotFoundException**
extends java.lang.Exception
implements java.io.Serializable

Constructor Index

● **UserNotFoundException**(String)

Constructors

● **UserNotFoundException**

```
 public UserNotFoundException(java.lang.String msg)
```

All Packages Class Hierarchy This Package Previous Next Index