

Slizaa 1.0.0.M3 Reference Guide

Table of Contents

Preface	1
1. Overview	1
1.1. What is Slizaa?	2
1.2. Why should I use Slizaa?	2
1.3. How does Slizaa work?	3
2. Getting Started	3
2.1. Prerequisites	4
2.2. Starting the Server	4
2.3. The Slizaa Console	4
2.4. Creating a Structure Database	5
2.5. Create a hierarchical graph	6
2.6. The Dependencies View	7
3. Using the Slizaa Shell	8
3.1. Basics	8
3.2. Admin Commands	8
3.3. Database Commands	9
3.4. Content Definition Commands	13
3.5. Hierarchical Graph Commands	15

Preface

This is the reference documentation of Slizaa, an interactive, web-based tool for analyzing and visualizing the dependencies of Java-based software systems. This guide covers all the functionality provided by Slizaa.

In case this guide doesn't answer all your questions, join the Slizaa [Google group](#) to get help. We also monitor the Slizaa tag on StackOverflow.

Did you find a typo or other error in this guide? Please let us know by opening an issue in the [Slizaa Documentation GitHub repository](#), or, better yet, help the community and send a pull request for fixing it!

This work is licensed under the [GNU Affero General Public License version 3.0](#).

1. Overview

1.1. What is Slizaa?

Slizaa is a web-based open source workbench for software dependency analysis.

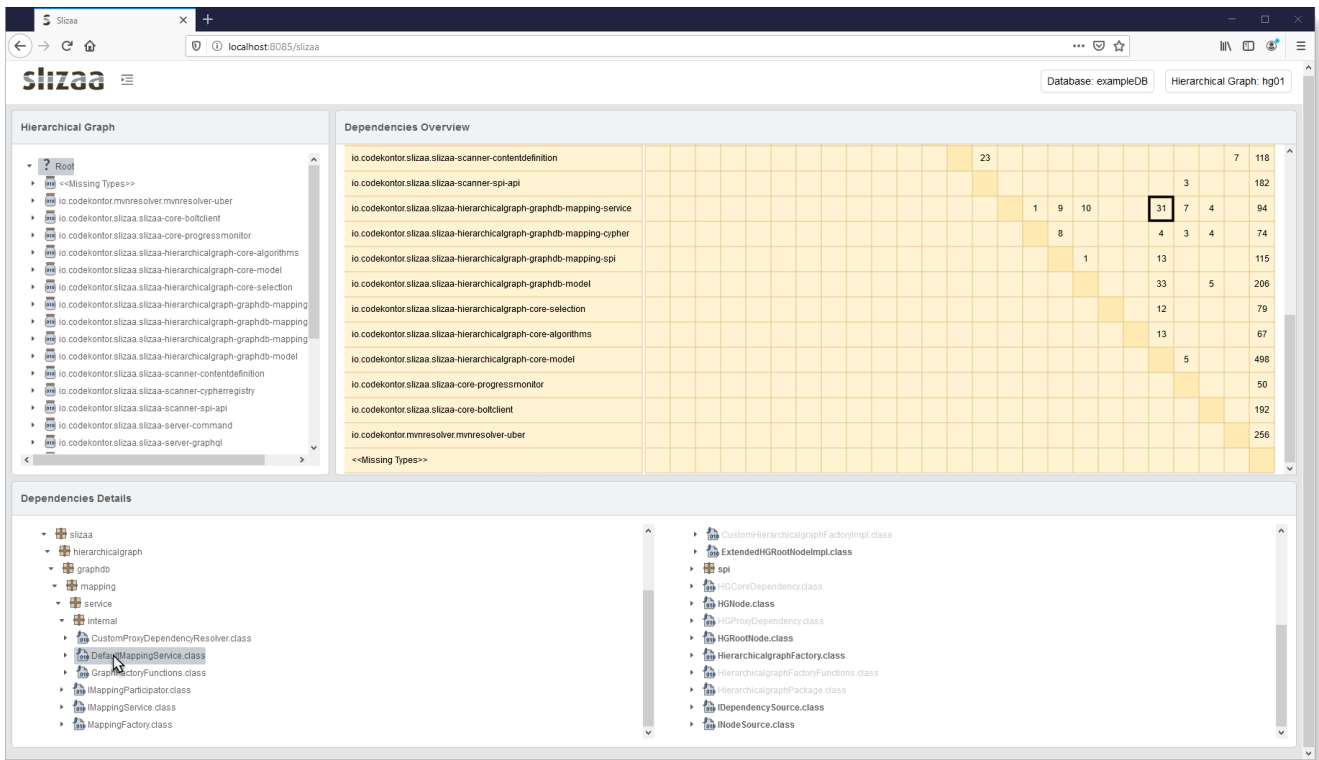


Figure 1. Slizaa Web Application

Slizaa provides a set of sophisticated tools to interactively explore and visualize the dependencies and the structure of java-based applications, e.g.

- a dependency structure matrix
- a powerful cross referencer [Not implemented in Slizaa 1.0.0.M3] and
- a dependency graph viewer [Not implemented in Slizaa 1.0.0.M3]

On top of that, Slizaa allows you to search for arbitrary patterns in your codebase using the simple but powerful Cypher query language directly from within the workbench [Not implemented in Slizaa 1.0.0.M3].

1.2. Why should I use Slizaa?

Maintainability of software is inversely proportional to the number of dependencies. Without continuous surveillance applications sooner or later become hard to maintain due to unwanted 'accidental' dependencies.

For continuous and efficient monitoring of your application's dependency structure, tool support is crucial. Slizaa provides an easy-to-use, web-based solution for dependency analysis and visualization.

1.3. How does Slizaa work?

Slizaa allows you to scan arbitrary (Java-based) applications and store their structural information in a graph database backend powered by Neo4j.

All these structural information is accessible directly from within Slizaa using the integrated Cypher query console [Not implemented in Slizaa 1.0.0.M3].

As searching for patterns in a flat graph can be a confusing and cluttering task, Slizaa provides customizable mappings to create hierarchical views of the stored software structure graph. A *hierarchical view* structures the elements of a software system (e.g., JARs, packages, types) in a hierarchy using 'is-contained-in' relationships. For software developers, hierarchical views are the natural way of navigating through software applications.

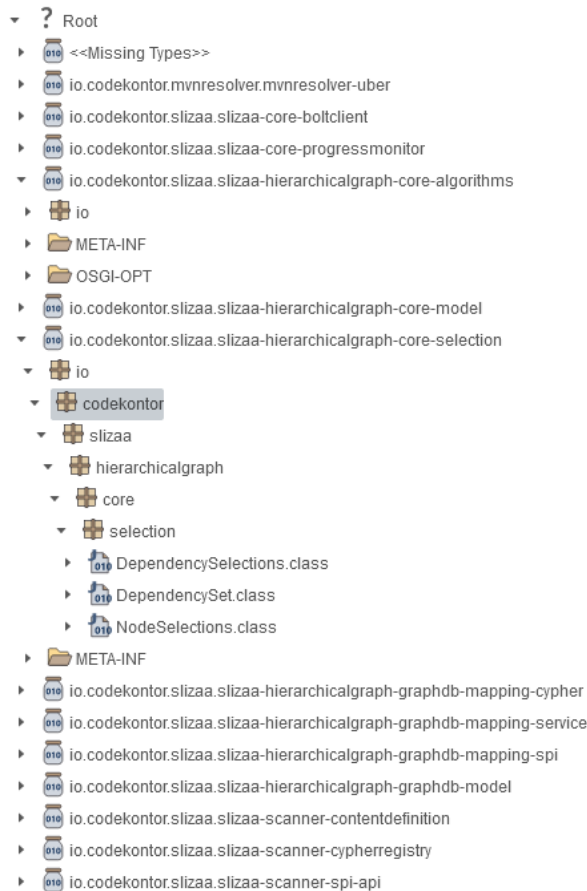


Figure 2. Hierarchical View

Using hierarchical views makes exploring your application’s architecture as easy as browsing code in your favorite IDE.

2. Getting Started

These instructions should help you get Slizaa up and running in 5 to 15 minutes.

2.1. Prerequisites

Slizaa requires a Java SE 8 (or higher) environment to run. Refer to the following URL for details on how to download and install Java SE 8 or higher: <http://www.oracle.com/technetwork/java/javase/>.

- Open a Web browser and access the following URL: <https://www.codekontor.io/slizaa/download/>
- Download the binary distribution that matches your system (zip for windows, tar.gz for unices)
- Extract the archive to a new folder on your hard drive (in this documentation we will reference this directory as `<SLIZAA_HOME>`)

2.2. Starting the Server

Open a command line console and change the directory to `<SLIZAA_HOME>`.

To start the server, run the following command:

```
java -jar slizaa-web-1.0.0.M3.jar
```

You should see the following information on the console:

```
      _--_
     /  ( )  \
    /  _  /  /  \  /  _  \  \
   (  _  ) /  /  /  \  /  /  /
  /  _  /  /  /  _  \  \  \  \
 /  _  /  /  /  _  \  \  \  \
```

```
Copyright © 2019 Code-Kontor GmbH and others (slizaa@codekontor.io)
```

```
[...]
```

```
slizaa:>
```

2.3. The Slizaa Console

You can now run your first command. To list all available commands, use the `help` command.

```

slizaa:>help
AVAILABLE COMMANDS

Built-In Commands
  clear: Clear the shell screen.
  exit, quit: Exit the shell.
  help: Display help about available commands.
  script: Read and execute commands from a file.
  stacktrace: Display the full stacktrace of the last error.

Slizaa Admin Commands
  gc: Run garbage collector.
  memUsage: Memory usage.

[...]

```

2.4. Creating a Structure Database

Before you can parse and analyze a software application, you have to create an instance of a structure database. A structure database is a place, where Slizaa stores the 'core' dependency information of a parsed software system. It is basically a graph database that is accessible via the BOLT protocol.

2.4.1. Listing all Structure Databases

To list all defined structure databases of a given Slizaa instance, execute the following command:

```

slizaa:>listDBs
No database configured.

```

2.4.2. Creating a new Structure Database

To create a new structure database instance, you can use the `createDB` command. You must provide a unique database name as the first parameter.

```

slizaa:>createDB exampleDB
+-----+-----+-----+-----+-----+-----+
|DatabaseId|State  |Port  |Content  |Hierarchical|Available Actions  |
|          |       |      |Definition|Graphs      |                   |
+-----+-----+-----+-----+-----+-----+
|exampleDB |INITIAL|59494|          |[]          |[setContentDefinitionProvi...|
|          |       |     |          |            |[delete]           |
+-----+-----+-----+-----+-----+-----+

```

2.4.3. Defining your Application Content

Once you have created a new structure database, you must specify the content that should be analyzed. Slizaa provides multiple *content definition providers* to provide a convenient way to specify which JARs or folders belong to your application.

In this tutorial, we use the **directory** content definition provider type to specify a folder on the local drive that contains JAR files that should be parsed. Make sure that

- the specified folder exists and
- the specified folder contains on or more JAR files.



If you are using this command on Windows in an ordinary windows shell, make sure that you escape the Backslashes that act as file separators (e.g., type **C:\\temp\\sl** instead of **C:\temp\sl**).

```
slizaa:>setContentDefinitionProvider exampleDB directory C:\tmp\sl
+-----+-----+-----+-----+-----+-----+
|DatabaseId|State   |Port |Content   |Hierarchical|Available Actions |
|          |        |     |Definition|Graphs      |                   |
+-----+-----+-----+-----+-----+-----+
|exampleDB |CONFIGURED|59494|directory |[]          |[parse,          |
|          |          |     |[C:\tmp\sl]|           |setContentDefinitionPr...|
|          |          |     |           |           |delete]         |
+-----+-----+-----+-----+-----+-----+
```

2.4.4. Parse the Specified Content

To parse the specified content and to populate the associated database with dependency structure information, use the following command:

```
slizaa:>parseDB exampleDB
+-----+-----+-----+-----+-----+-----+
|DatabaseId|State   |Port |Content Definition|Hierarchical   |Available   |
|          |        |     |                  |Graphs        |Actions     |
+-----+-----+-----+-----+-----+-----+
|exampleDB |RUNNING |59494|directory        |[]            |[stop, delete]|
|          |        |     |[C:\tmp\sl]     |              |               |
+-----+-----+-----+-----+-----+-----+
```

2.5. Create a hierarchical graph

Once you have populated the internal graph database, you can create a hierarchical view of the parsed software system by using the **createHierarchicalGraph** command, as shown below.

```
slizaa:>createHierarchicalGraph exampleDB hg01
```

DatabaseId	State	Port	Content Definition	Hierarchical Graphs	Available Actions
exampleDB	RUNNING	59494	directory [C:\tmp\sl]	[hg01]	[stop, delete]

2.6. The Dependencies View

To inspect and analyze the parsed software system,

- open a Web browser, and access the following URL: <https://localhost:8085/slizaa>. The Slizaa web application automatically opens with the *Dependencies View*.

Select the desired database in the database selector in the upper right corner, as well as the desired hierarchical graph.

The screenshot shows the Slizaa web application interface. At the top right, there are dropdowns for 'Database: exampleDB' and 'Hierarchical Graph: hg01'. The main area is divided into three sections:

- Hierarchical Graph:** A tree view on the left showing a hierarchy of packages under 'slizaa', including 'graphdb', 'mapping', 'service', and 'internal'.
- Dependencies Overview:** A matrix in the center showing dependencies between packages. The matrix has columns for each package and rows for each package. The values in the matrix represent the number of dependencies. For example, 'io.codekontor.slizaa.scanner.contentdefinition' has 23 dependencies, and 'io.codekontor.slizaa.scanner.spi' has 182 dependencies.
- Dependencies Details:** A list of dependencies on the right, showing the source class for each dependency. For example, 'CustomHierarchicalGraphFactoryImpl.class' is a dependency of 'ExtendedHRootNodeImpl.class'.

Figure 3. The Dependencies View

2.6.1. The Hierarchical Graph

The Dependencies View visualizes the hierarchical representation of the parsed system in the upper left part (*Hierarchical Graph*).

2.6.2. Dependencies Overview

The *Dependencies Overview* visualizes the dependencies between the children of the selected element in a Dependency Structure Matrix. You can select an arbitrary node in the hierarchical graph to inspect the dependencies of its children in the Dependencies Overview.

2.6.3. Dependencies Details

You can inspect details of dependencies in the *Dependencies Details* view in the lower part of the window. To do so, select a cell in the Dependency Structure Matrix by clicking on it.

3. Using the Slizaa Shell

3.1. Basics

To list all available commands, use the `help` command.

```
slizaa:>help
AVAILABLE COMMANDS

Built-In Commands
  clear: Clear the shell screen.
  exit, quit: Exit the shell.
  help: Display help about available commands.
  script: Read and execute commands from a file.
  stacktrace: Display the full stacktrace of the last error.

Slizaa Admin Commands
  gc: Run garbage collector.
  memUsage: Memory usage.

[...]
```

3.2. Admin Commands

The admin command group contains several commands to monitor and control the overall state of the Slizaa server instance.

```
slizaa:>help
AVAILABLE COMMANDS

[...]
```

Slizaa Admin Commands

```
  memUsage: Memory usage.
  gc: Run garbage collector.

[...]
```

3.2.1. Show Memory Usage

You can show the current memory usage of a Slizaa server instance using the command `memUsage`. The command prints the memory currently used as well as the total memory on the console.


```
slizaa:>memUsage
Current Memory Usage:
Used Memory: 161 MB
Total Memory: 1021 MB
```

The `memUsage` command is defined as follows:

```
slizaa:>help memUsage
NAME
    memUsage - Memory usage.

SYNOPSIS
    memUsage
```

3.2.2. Run the Garbage Collector

Although it is usually not necessary, you can manually run the garbage collector using the command `gc`.

```
slizaa:>gc
Current Memory Usage:
Used Memory: 102 MB
Total Memory: 1135 MB
```

The `gc` command is defined as follows:

```
slizaa:>help gc
NAME
    gc - Run garbage collector.

SYNOPSIS
    gc
```

3.3. Database Commands

The commands in the database command group allow you to inspect and manipulate the structure databases in a Slizaa instance.

3.3.1. List all configured databases

You can use the command `listDBs` to list the databases configured in a Slizaa instance.

```
slizaa:>listDBs
No database configured.
```

The `listDBs` command is defined as follows:

```
slizaa:>help listDBs
NAME
  listDBs - List all configured databases.

SYNOPSIS
  listDBs
```

3.3.2. Create a new database

To create a new database instance, use the `createDB` command.

```
slizaa:>createDB exampleDB
+-----+-----+-----+-----+-----+-----+
|DatabaseId|State  |Port |Content  |Hierarchical|Available Actions  |
|          |       |     |Definition|Graphs      |                   |
+-----+-----+-----+-----+-----+-----+
|exampleDB |INITIAL|30166|          |[]          |[setContentDefinitionProvi...|
|          |       |     |          |            |[delete]           |
+-----+-----+-----+-----+-----+-----+
```

The `createDB` command is defined as follows:

```
slizaa:>help createDB
NAME
  createDB - Create a new database.

SYNOPSIS
  createDB [-d] string

OPTIONS
  -d or --databaseId string
    The identifier of the database to create.
    [Mandatory]
```

3.3.3. Start the specified database

To start an existing database instance, use the command `startDB`:

```
slizaa:>startDB exampleDB
```

```
+-----+-----+-----+-----+-----+-----+
|DatabaseId|State  |Port |Content Definition|Hierarchical |Available  |
|          |      |    |                |Graphs      |Actions   |
+-----+-----+-----+-----+-----+-----+
|exampleDB |RUNNING|30166|directory      |[]          |[stop, delete]|
|          |      |    |[C:\tmp\s1]   |           |[             ]|
+-----+-----+-----+-----+-----+-----+
```

The `startDB` command is defined as follows:

```
slizaa:>help startDB
```

NAME

startDB - Start the specified database.

SYNOPSIS

startDB [-d] string

OPTIONS

-d or --databaseId string

The identifier of the database to start.

[Mandatory]

3.3.4. Stop the specified database.

To stop an existing database instance, use the command `stopDB`.

```
slizaa:>stopDB exampleDB
```

```
+-----+-----+-----+-----+-----+-----+
|DatabaseId|State  |Port |Content  |Hierarchical|Available Actions  |
|          |      |    |Definition|Graphs      |                  |
+-----+-----+-----+-----+-----+-----+
|exampleDB |NOT_RUNNING|30166|directory  |[]          |[setContentDefinition...|
|          |          |    |[C:\tmp\s1]|           |[parse, start, delete]|
+-----+-----+-----+-----+-----+-----+
```

The `stopDB` command is defined as follows:

```

slizaa:>help stopDB
NAME
  stopDB - Stop the specified database.

SYNOPSIS
  stopDB [-d] string

OPTIONS
  -d or --databaseId string
    The identifier of the database to stop.
    [Mandatory]

```

3.3.5. Delete an existing database

To delete an existing database instance, use the `deleteDB` command.

```

slizaa:>deleteDB exampleDB
No database configured.

```

The `deleteDB` command is defined as follows:

```

slizaa:>help deleteDB
NAME
  deleteDB - Delete an existing database.

SYNOPSIS
  deleteDB [-d] string

OPTIONS
  -d or --databaseId string
    The identifier of the database to delete.
    [Mandatory]

```

3.3.6. Parse the defined content

To parse the specified content for a database, use the `parseDB` command.

```

slizaa:>parseDB exampleDB
+-----+-----+-----+-----+-----+-----+
|DatabaseId|State  |Port |Content Definition|Hierarchical  |Available  |
|          |       |     |                  |Graphs        |Actions    |
+-----+-----+-----+-----+-----+-----+
|exampleDB |RUNNING|30166|directory        |[[]           |[stop, delete]|
|          |       |     |[C:\tmp\s1]     |              |              |
+-----+-----+-----+-----+-----+-----+

```

The `parseDB` command is defined as follows:

```
slizaa:>help parseDB
NAME
  parseDB - Parse the defined content.

SYNOPSIS
  parseDB [-d] string

OPTIONS
  -d or --databaseId string
    The identifier of the database to parse.
    [Mandatory]
```

3.4. Content Definition Commands

You can specify the content of an application that should be analyzed using *content definition commands*. The *content* of an application consists of the JARs and directories that contain classes belonging to your application.

Instead of listing all JARs and directory "manually", you usually use a so-called *content definition providers* to specify what should be parsed. Content definition providers implement more convenient ways to define your content (e.g. by simply specifying a directory that contains JARs and directories).

In Slizaa 1.0.0.M3 there are two different content definition provider you can choose of:

- **directory**: Using the `directory` content definition provider you can specify an arbitrary folder on your local drive that contains the JAR files that Slizaa should parse.
- **mvn**: Using the `mvn` content definition provider you can specify a list of maven artifact that Slizaa should parse. Slizaa automatically will download the specified artifacts during the parsing process.

3.4.1. Specify the content to be analyzed.

You can specify your applications content using the `setContentDefinitionProvider` command as shown below:

```
slizaa:>setContentDefinitionProvider exampleDB directory C:\tmp\sl
+-----+-----+-----+-----+-----+-----+
|DatabaseId|State   |Port |Content  |Hierarchical|Available Actions  |
|          |        |     |Definition|Graphs      |                    |
+-----+-----+-----+-----+-----+-----+
|exampleDB |CONFIGURED|30166|directory |[]          |[parse,            |
|          |          |     |[C:\tmp\sl]|            |setContentDefinitionPr...|
|          |          |     |          |            |delete]            |
+-----+-----+-----+-----+-----+-----+
```

The `setContentDefinitionProvider` command is defined as follows:

```
slizaa:>help setContentDefinitionProvider
NAME
  setContentDefinitionProvider - Define the content that should be analyzed.

SYNOPSIS
  setContentDefinitionProvider [-d] string [-f] string [-c] string

OPTIONS
  -d or --databaseId string
      [Mandatory]

  -f or --factoryId string
      [Mandatory]

  -c or --contentDefinition string
      [Mandatory]
```

3.4.2. List the available Content Definition Types

You can list the available content definition provider types using the `listContentDefinitionProviderFactories` command:

```
slizaa:>listContentDefinitionProviderFactories
Content Definition Provider Factories:
- directory (io.codekontor.slizaa.scanner.contentdefinition.DirectoryBasedCo...
- mvn (io.codekontor.slizaa.scanner.contentdefinition.MvnBasedContentDefinit...
```

The `listContentDefinitionProviderFactories` command is defined as follows:

```
slizaa:>help listContentDefinitionProviderFactories
NAME
  listContentDefinitionProviderFactories - List available content definition
  provider factories.

SYNOPSIS
  listContentDefinitionProviderFactories
```

3.4.3. Show Resolved Content Definitions

To simulate the resolving process and fetch a list of all JARs and directories that are defined by a content definition use the `showResolvedContentDefinitions` command:

```
slizaa:>showResolvedContentDefinitions exampleDB
Content Definitions:
- io.codekontor.mvnresolver.mvnresolver-uber_1.0.0.RC1
  - C:\tmp\sl\mvnresolver-uber-1.0.0.RC1.jar
- io.codekontor.slizaa.slizaa-core-boltclient_1.0.0.M3
  - C:\tmp\sl\slizaa-core-boltclient-1.0.0.M3.jar
- io.codekontor.slizaa.slizaa-core-progressmonitor_1.0.0.M3
  - C:\tmp\sl\slizaa-core-progressmonitor-1.0.0.M3.jar
- io.codekontor.slizaa.slizaa-hierarchicalgraph-core-algorithms_1.0.0.M3
  - C:\tmp\sl\slizaa-hierarchicalgraph-core-algorithms-1.0.0.M3.jar
- io.codekontor.slizaa.slizaa-hierarchicalgraph-core-model_1.0.0.M3

[...]
```

The `showResolvedContentDefinitions` command is defined as follows:

```
slizaa:>help showResolvedContentDefinitions
NAME
  showResolvedContentDefinitions - List the content definitions for the specified
  database.

SYNOPSIS
  showResolvedContentDefinitions [-d] string [-f]

OPTIONS
  -d or --databaseId string
      [Mandatory]

  -f or --showFiles
      [Optional, default = false]
```

3.5. Hierarchical Graph Commands

3.5.1. Create Hierarchical Graph

You can create a new hierarchical graph mapping for a database using the `createHierarchicalGraph` command.

```
slizaa:>createHierarchicalGraph exampleDB hg01
```

DatabaseId	State	Port	Content Definition	Hierarchical Graphs	Available Actions
exampleDB	RUNNING	30166	directory [C:\tmp\sl]	[hg01]	[stop, delete]

The `createHierarchicalGraph` command is defined as follows:

```
slizaa:>help createHierarchicalGraph
```

NAME
 createHierarchicalGraph - Create a new hierarchical graph.

SYNOPSIS
 createHierarchicalGraph [-d] string [-h] string

OPTIONS
 -d or --databaseId string
 [Mandatory]

-h or --hierarchicalGraphId string
 [Mandatory]

3.5.2. Delete Hierarchical Graph

```
slizaa:>deleteHierarchicalGraph exampleDB hg01
```

Can not execute command.
 The specified hierarchical graph 'hg01' does not exist.

The `createHierarchicalGraph` command is defined as follows:


```
slizaa:>help deleteHierarchicalGraph
```

NAME

deleteHierarchicalGraph - Delete an existing hierarchical graph.

SYNOPSIS

deleteHierarchicalGraph [-d] string [-h] string

OPTIONS

-d or --databaseId string

[Mandatory]

-h or --hierarchicalGraphId string

[Mandatory]