TruffleHog

Generated by Doxygen 1.8.10

Wed Jan 27 2016 01:18:17

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 de.fraunhofer.iosb.trufflehog.service.datalogging.CommandCompressor Class Reference

Public Member Functions

- List< ICommand > compressCommands (List< ICommand > commands)

3.1.1 Detailed Description

The CommandCompressor creates a compacted list of all commands it received by using the Command↩ Compressor. For example, if there are 50 consecutive commands in the list that all increment the same counter by 1, these 50 commands are packaged into 1 command that increments the counter by 50 and then returned as a single command.

3.1.2 Member Function Documentation

3.1.2.1 List<ICommand> de.fraunhofer.iosb.trufflehog.service.datalogging.CommandCompressor.compressCommands (List< ICommand > *commands*)

Creates a compacted list of all commands it received by using the CommandCompressor. For example, if there are 50 consecutive commands in the list that all increment the same counter by 1, these 50 commands are packaged into 1 command that increments the counter by 50 and then returned as a single command.

Parameters

commands	The commands to compress.

Returns

The compressed commands.

3.2 de.fraunhofer.iosb.trufflehog.service.datalogging.CommandLogger Class Reference

Public Member Functions

- CommandLogger ()
- void addCommand (ICommand command)
- List< ICommand > createCommandLog ()

3.2.1 Detailed Description

The CommandLogger takes a list of commands, packages them into a more "dense" list through the CommandCompressor. That means if for instance there are 50 consecutive commands that all increment the same counter by 1, these 50 commands are packaged into 1 command that increments the counter by 50. This compacted list is returned to the DataLogLoadService where the command list is packaged into a DataLog.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 de.fraunhofer.iosb.trufflehog.service.datalogging.CommandLogger.CommandLogger ( )

Creates a new CommandLogger object.

3.2.3 Member Function Documentation

3.2.3.1 void de.fraunhofer.iosb.trufflehog.service.datalogging.CommandLogger.addCommand (ICommand *command*)

Adds a command to the internal command list. The list will be taken by createCommandLog() and then cleared.

Parameters

command	The command to add to the list that is to be processed.

3.2.3.2 List<ICommand> de.fraunhofer.iosb.trufflehog.service.datalogging.CommandLogger.createCommandLog ( )

Creates a compacted list of all commands it received by using the CommandCompressor. For example, if there are 50 consecutive commands in the list that all increment the same counter by 1, these 50 commands are packaged into 1 command that increments the counter by 50 and then returned as a single command.

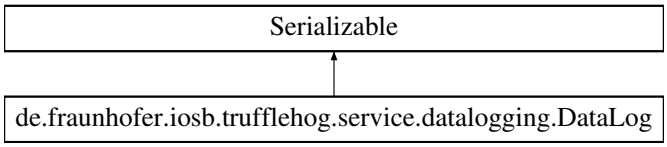
Once executed, all storred commands are deleted again.

Returns

A list of compacted commands it received.

3.3 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLog Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.service.datalogging.DataLog:



Public Member Functions

- DataLog (INetworkGraph snapshotGraph, List< ICommand > commands)
- INetworkGraph getGraphSnapshot ()
- List< ICommand > getCommands ()
- Instant getStartInstant ()
- Instant getEndInstant ()

3.3.1 Detailed Description

A DataLog object contains the snapshot of a given graph, a list of all commands that were executed for the next X seconds following that snapshot and a timestamp of when the DataLog starts to where it ends.

DataLog objects are used to recreate a graph that already occurred. They capture all necessary information to display the full graph the way it was at any given point in time. This is done by applying the commands stored in it back on the snapshot until the end of the DataLog is reached, at which point the commands of the next data log are applied and so on.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLog.DataLog (INetworkGraph *snapshotGraph,* List< ICommand > *commands*)

Creates a new DataLog object containing the given graph snapshot and the list of commands.

Parameters

snapshotGraph	The snapshot of the graph to include in the data log.
commands	The list of commands to include in the data log.

3.3.3 Member Function Documentation

3.3.3.1 List<ICommand> de.fraunhofer.iosb.trufflehog.service.datalogging.DataLog.getCommands ( )

Gets the command list contained in this DataLog object.

Returns

The command list contained in this DataLog object.

3.3.3.2 Instant de.fraunhofer.iosb.trufflehog.service.datalogging.DataLog.getEndInstant ( )

Gets the point in time of when this DataLog object stops to contain graph data.

Returns

The point in time of when this DataLog object stops to contain graph data.

3.3.3.3 INetworkGraph de.fraunhofer.iosb.trufflehog.service.datalogging.DataLog.getGraphSnapshot ( )

Gets the graph snapshot contained in this DataLog object.

Returns

The graph snapshot contained in this DataLog object.

3.3.3.4 Instant de.fraunhofer.iosb.trufflehog.service.datalogging.DataLog.getStartInstant ( )

Gets the point in time of when this DataLog object starts to contain graph data.

Returns

The point in time of when this DataLog object starts to contain graph data.

3.4 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogger Class Reference

Public Member Functions

- DataLogger ()
- DataLog createDataLog (INetworkGraph snapshotGraph, List< ICommand > commands)
- void saveDataLog (DataLog log)
- void outputDataLog (DataLog log)

3.4.1 Detailed Description

The DataLogger manages the saving of the graph structure. It creates DataLog objects which are serialized and saved. From these DataLog objects, the entire graph can be recreated from a certain point in time.

The DataLogger saves the state of the graph through two things. It takes snapshots of the graph every X seconds through the SnapshotLogger and then saves all commands that occurred from that point on until the next snapshot is taken through the CommandLogger. Together these snapshots with the list of commands form a DataLog. When blocks of DataLogs that were saved consecutively are loaded back into memory, they can be used to reconstruct the graph by taking the snapshot as the original graph and then applying all commands that occurred back on the graph, in the order and interval they occurred. This is how old graphs can be viewed.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogger.DataLogger ( )

Creates a new DataLogger.

3.4.3 Member Function Documentation

3.4.3.1 DataLog de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogger.createDataLog (INetworkGraph *snapshotGraph,* List< ICommand > *commands*)

Creates a new DataLog object based on the snapshot of a graph given and the list of commands that were executed for the next X seconds after the snapshot was taken.

Parameters

snapshotGraph	The snapshot of the graph to include in the data log.
commands	The list of commands to include in the data log.

Returns

A new DataLog object that is serializable and that contains the graph snapshot and the list of commands passed to the method.

3.4.3.2 void de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogger.outputDataLog (DataLog *log*)

Outputs the raw data of a DataLog object to the view, so that the user can see all kinds of information about the graph its content at a given point in time.

Parameters

log	The DataLog object whose internal data should be shown on screen.

3.4.3.3 void de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogger.saveDataLog (DataLog *log*)

Saves the DataLog object given on the hard drive so that it can be retrieved later.

Parameters

log	The data log to save on the hard drive.

3.5 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoader Class Reference

Public Member Functions

- DataLogLoader ()
- void loadData (Instant instant)
- DataLog getData (Instant instant) throws MissingResourceException

3.5.1 Detailed Description

The DataLogLoader loads DataLogs from the hard drive into memory, so that when the view needs them to display the old graph, it is at the view's disposal.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoader.DataLogLoader ( )

Creates a new DataLogLoader object.

3.5.3 Member Function Documentation

3.5.3.1 DataLog de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoader.getData (Instant *instant*) throws MissingResourceException

Gets the DataLog object closest to the given time instant.

Condition: A DataLog reasonably close in time to the given instant must be in memory, else a MissingResource↩ Exception is thrown.

Parameters

instant	The time instant that should be used to get the DataLog object. The DataLog object closest to the time instant will be returned.

Returns

The DataLog object closest in time to the given time instant.

Exceptions

MissingResourceException	Thrown when no DataLog reasonably close in time to the given instant is found in memory.

3.5.3.2 void de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoader.loadData (Instant *instant*)

Loads a batch of DataLogs closest to the given time instant into memory so that they are ready when needed.

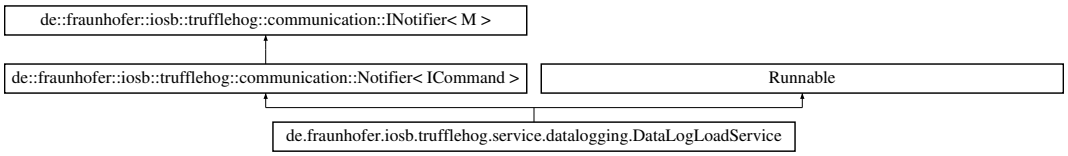
This method should not be confused with getData(Instant). getData(Instant) gets the desired DataLog object from the already loaded DataLog objects and does not load any DataLog itself.

Parameters

instant	The time instant that should be used to load the DataLog objects. The DataLog objects closest to the time instant will be loaded.

3.6 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoadService Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoadService:



Public Member Functions

- DataLogLoadService ()
- void play ()
- void pause ()
- void jumpToInstant (Instant instant)
- void load (Instant instant)
- void run ()

3.6.1 Detailed Description

The DataLogLoadService controls the playback of the playback graph (the old graph, generated from the

`DataLog`

s). It runs in its own thread, however most of its methods are called in other threads where the actual playback is controlled. Instead the DataLogLoadService makes sure there is always data to playback in the background by loading and buffering DataLog objects, and dispatching the commands of the DataLog objects. (The DataLog↩ LoadService can do this since it is also a Notifier).

3.6.2 Constructor & Destructor Documentation

3.6.2.1 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoadService.DataLogLoadService ( )

Creates a new DataLogLoadService object.

3.6.3 Member Function Documentation

3.6.3.1 void de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoadService.jumpToInstant (Instant *instant*)

The current playback of the DataLogs will jump to the location in time closest to the given instant, if the playback is currently active.

Parameters

instant	The instant in time to which to jump in the playback, if it is active.

3.6.3.2 void de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoadService.load (Instant *instant*)

Loads a batch of DataLogs closest to the given time instant into memory so that they are ready when needed. It does this through the DataLogLoader.

Parameters

instant	The time instant that should be used to load the DataLog objects. The DataLog objects closest to the time instant will be loaded.

3.6.3.3 void de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoadService.pause ( )

Pauses the playback of the DataLogs if it was in progress.

3.6.3.4 void de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoadService.play ( )

Starts the playback of the DataLogs. The graph snapshot will be loaded and the subsequent commands will be executed on that snapshot.

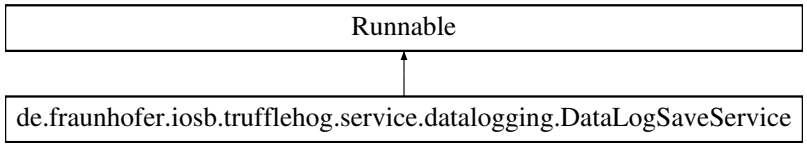
3.6.3.5 void de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogLoadService.run ( )

Starts the data log load service.

The DataLogLoadService runs in its own thread to buffer DataLog objects in the background so that when the graph playback function is activated, the old graph can immediately be started. It also notifies the playback executor with the old commands found in the DataLog objects so that the executor can update the playback graph.

3.7 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogSaveService Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogSaveService:



Public Member Functions

- DataLogSaveService ()
- void run ()

3.7.1 Detailed Description

The DataLogSaveService saves the current graph state so that it can be fully reconstructed at a later date. It does this by creating DataLog objects at a fixed time interval (each DataLog object covers the data for this time interval). Thus the DataLogSaveService runs in its own thread.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogSaveService.DataLogSaveService ( )

Creates a new DataLogSaveService object.

3.7.3 Member Function Documentation

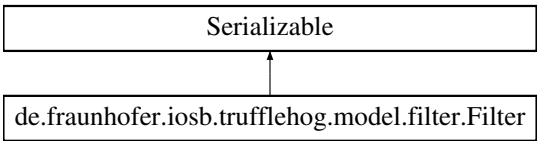
3.7.3.1 void de.fraunhofer.iosb.trufflehog.service.datalogging.DataLogSaveService.run ( )

Starts the data log save service.

The DataLogSaveService saves the current graph state so that it can be fully reconstructed at a later date. It does this by creating DataLog objects at a fixed time interval (each DataLog object covers the data for this time interval). Thus the DataLogSaveService runs in its own thread.

3.8 de.fraunhofer.iosb.trufflehog.model.filter.Filter Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.filter.Filter:



Public Member Functions

- void addIPRegex (String ex)
- void addMACRegex (String ex)
- void **addNameRegex** (String ex)

3.8.1 Detailed Description

This class is used to store filtering options such as white- and blacklists.

3.8.2 Member Function Documentation

3.8.2.1 void de.fraunhofer.iosb.trufflehog.model.filter.Filter.addIPRegex (String *ex*)

p> Adds an MAC-rule to the regex list.

Parameters

ex	Expression to add

3.8.2.2 void de.fraunhofer.iosb.trufflehog.model.filter.Filter.addMACRegex (String *ex*)

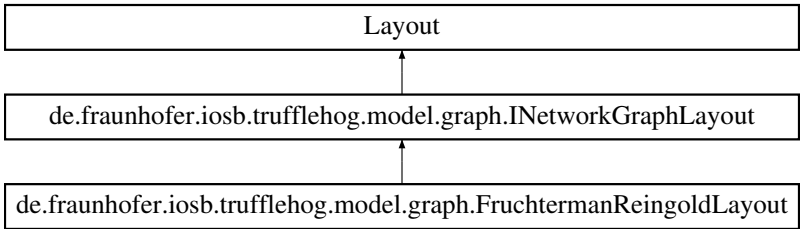
p> Adds an Name-rule to the regex list.

Parameters

ex	Expression to add

3.9 de.fraunhofer.iosb.trufflehog.model.graph.FruchtermanReingoldLayout Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.FruchtermanReingoldLayout:



Public Member Functions

- void FruchtermanReingoldLayout (INetworkGraph graph)
- void **initialize** ()
- void **setGraph** (Graph graph)
- Graph **getGraph** ()
- void **reset** ()
- void **setSize** (Dimension d)
- Dimension **getSize** ()
- void **lock** (Object o, boolean state)
- boolean **isLocked** (Object o)
- void **setLocation** (Object o, Point2D location)
- void **setInitializer** (Transformer initializer)
- Object **transform** (Object o)

3.9.1 Detailed Description

Uses the Fruchterman-Reingold-algorithm from the jung library to present the INetworkGraph.

3.9.2 Constructor & Destructor Documentation

3.9.2.1 void de.fraunhofer.iosb.trufflehog.model.graph.FruchtermanReingoldLayout.FruchtermanReingoldLayout (INetworkGraph *graph*)

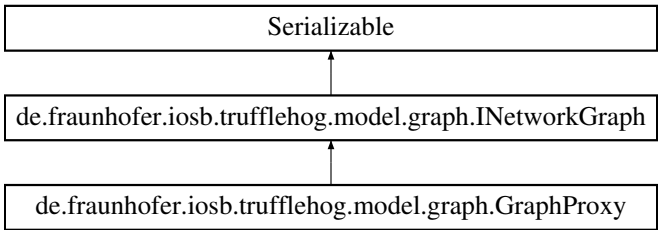
Creates a new layout to present a given INetworkGraph.

Parameters

graph	INetworkGraph to be drawn.

3.10 de.fraunhofer.iosb.trufflehog.model.graph.GraphProxy Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.GraphProxy:



Public Member Functions

- void **setGraph** (INetworkGraph graph)
- void addNetworkNode (NetworkNode node)
- void addNetworkEdge (NetworkNode from, NetworkNode to)

3.10.1 Detailed Description

Graph deputy used to encapsulate the graph and thus make it interchangeable without having to update graph references of other objects.

3.10.2 Member Function Documentation

3.10.2.1 void de.fraunhofer.iosb.trufflehog.model.graph.GraphProxy.addNetworkEdge (NetworkNode *from,* NetworkNode *to*)

Implements de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraph.

3.10.2.2 void de.fraunhofer.iosb.trufflehog.model.graph.GraphProxy.addNetworkNode (NetworkNode *node*)

Adds a new NetworkNode to the graph to represent a new device.

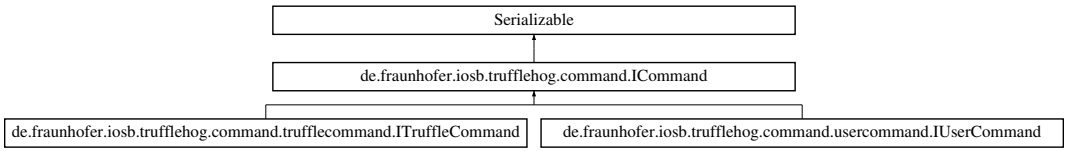
Parameters

node	NetworkNode to add p> Adds a new NetworkEdge to the graph to represent a new communication protocol used by two nodes.
from	NetworkNode sending packages
to	NetworkNode receiving packages

Implements de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraph.

3.11 de.fraunhofer.iosb.trufflehog.command.ICommand Interface Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.command.ICommand:

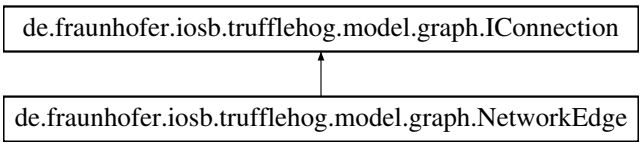


Public Member Functions

- void **execute** ()

3.12 de.fraunhofer.iosb.trufflehog.model.graph.IConnection Interface Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.IConnection:

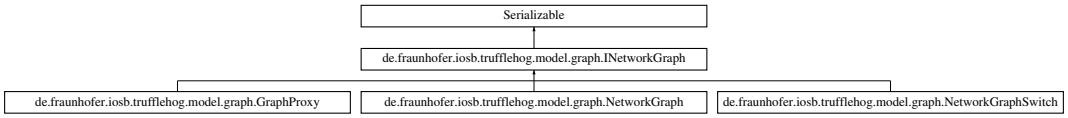


3.12.1 Detailed Description

Interface used to represent communication relations of nodes in the graph.

3.13 de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraph Interface Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraph:



Public Member Functions

- void addNetworkNode (NetworkNode node)
- void **addNetworkEdge** (NetworkNode from, NetworkNode to)

3.13.1 Detailed Description

Interface for the network graph.

3.13.2 Member Function Documentation

3.13.2.1 void de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraph.addNetworkNode (NetworkNode *node*)

Adds a new NetworkNode to the graph to represent a new device.

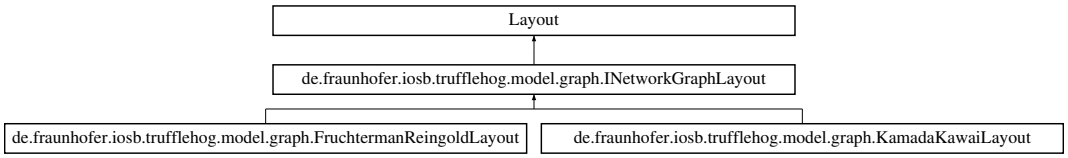
Parameters

node	NetworkNode to add p> Adds a new NetworkEdge to the graph to represent a new communication protocol used by two nodes.
from	NetworkNode sending packages
to	NetworkNode receiving packages

Implemented in de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraphSwitch, de.fraunhofer.iosb.trufflehog.↩ model.graph.NetworkGraph, and de.fraunhofer.iosb.trufflehog.model.graph.GraphProxy.

3.14 de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraphLayout Interface Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraphLayout:

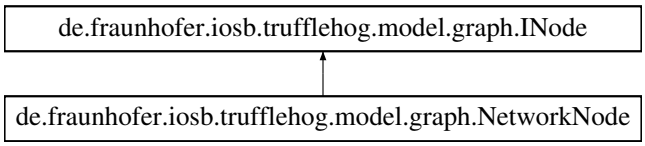


3.14.1 Detailed Description

Interface to exchange graph drawing algorithms.

3.15 de.fraunhofer.iosb.trufflehog.model.graph.INode Interface Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.INode:

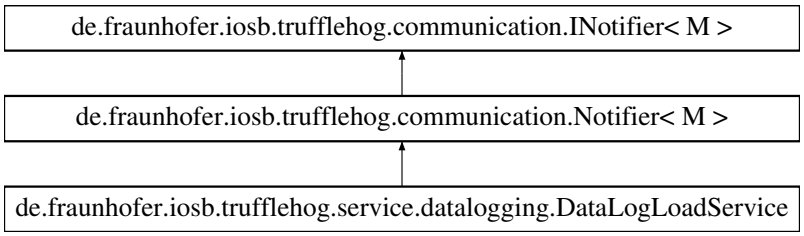


3.15.1 Detailed Description

Interface used to represent nodes in the graph.

3.16 de.fraunhofer.iosb.trufflehog.communication.INotifier< M > Interface Template Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.communication.INotifier< M >:



Public Member Functions

- void addListener (Listener listener)
- void removeListener (Listener listener)
- void notifyListeners (M message)

3.16.1 Detailed Description

The INotifier interface works together with Notifier and Listener to create the main communication method between threads in TruffleHog. This communication method is a variation of the observer design pattern where messages sent from the subject to the observer include a parameter of type M. The INotifier is the subject in this case. IListeners register with the INotifier and can then be notified through the notifyListeners method. Unlike in the classic observer pattern, the notifyListeners method offers the possibility to pass along a parameter of type M.

The INotifier is the abstraction from the Notifier. If a class needs to be notified and cannot extend it, it should implement INotifier and its methods should be delegated to the actual Notifier class.

Parameters

<M>	The type of message to receive.

3.16.2 Member Function Documentation

3.16.2.1 void de.fraunhofer.iosb.trufflehog.communication.INotifier< M >.addListener (Listener *listener*)

Register an IListener with this INotifier. This IListener will then be notified on the notifyListeners(M) method call.

Parameters

listener	The IListeners to register with this INotifier.

Implemented in de.fraunhofer.iosb.trufflehog.communication.Notifier< M >.

3.16.2.2 void de.fraunhofer.iosb.trufflehog.communication.INotifier< M >.notifyListeners (M *message*)

Notifies all IListeners that are registered with this INotifier. It sends along a message of type M

Parameters

message	The message to send along with the notification to each IListener.

Implemented in de.fraunhofer.iosb.trufflehog.communication.Notifier< M >.

3.16.2.3 void de.fraunhofer.iosb.trufflehog.communication.INotifier< M >.removeListener (Listener *listener*)

Removes an IListener from this INotifier. This IListener will then not be notified anymore on the notifyListeners(M) method call.

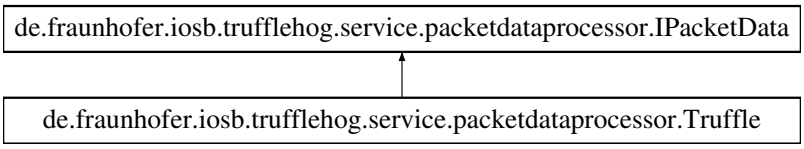
Parameters

listener	The IListeners to register with this INotifier.

Implemented in de.fraunhofer.iosb.trufflehog.communication.Notifier< M >.

3.17 de.fraunhofer.iosb.trufflehog.service.packetdataprocessor.IPacketData Interface Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.service.packetdataprocessor.IPacketData:



Public Member Functions

- < T > T getAttribute (Class< T > attributeType, String attributeIdentifier)

3.17.1 Detailed Description

This Interface provides the default methods for any incoming packet data which is received by e.g. a TruffleReceiver and put into a class (e.g a truffle) that implements this interface.

Author

Mr. X

Version

0.0

3.17.2 Member Function Documentation

3.17.2.1 <T> T de.fraunhofer.iosb.trufflehog.service.packetdataprocessor.IPacketData.getAttribute (Class< T > *attributeType*, String *attributeIdentifier*)

This method gets an attribute of the packet data object using the specified attribute name.

The function should always return a valid element of the specified type. Each implementation has to make sure this rule is never violated.

In case no element was found under the specified identifier, null is returned.

Parameters

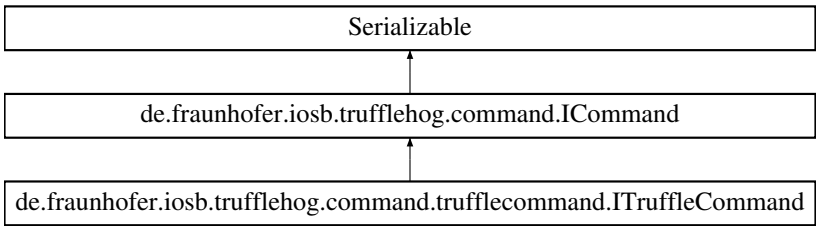
attributeType	The type of attribute that is supposed to be retrieved, for example Integer.class
attributeIdentifier	The string identifier of the attribute that should be retrieved.
<T>	The type of the attribute that is retrieved.

Returns

The value of the attribute or null if nothing was found under the specified identifier

3.18 de.fraunhofer.iosb.trufflehog.command.trufflecommand.ITruffleCommand Interface Reference

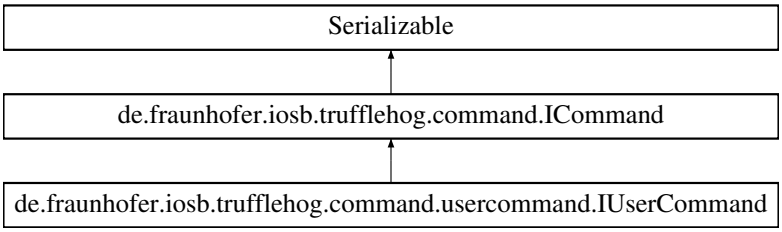
Inheritance diagram for de.fraunhofer.iosb.trufflehog.command.trufflecommand.ITruffleCommand:



Additional Inherited Members

3.19 de.fraunhofer.iosb.trufflehog.command.usercommand.IUserCommand Interface Reference

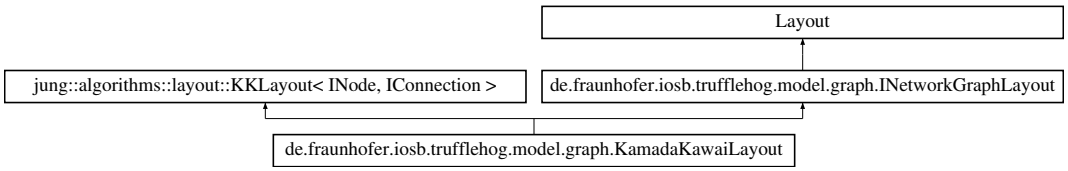
Inheritance diagram for de.fraunhofer.iosb.trufflehog.command.usercommand.IUserCommand:



Additional Inherited Members

3.20 de.fraunhofer.iosb.trufflehog.model.graph.KamadaKawaiLayout Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.KamadaKawaiLayout:



Public Member Functions

- void KamadaKawaiLayout (INetworkGraph graph)

3.20.1 Detailed Description

Uses the Kamada-Kawai-algorithm from the jung library to present the INetworkGraph.

3.20.2 Constructor & Destructor Documentation

3.20.2.1 void de.fraunhofer.iosb.trufflehog.model.graph.KamadaKawaiLayout.KamadaKawaiLayout (INetworkGraph *graph*)

Creates a new layout to present a given INetworkGraph.

Parameters

graph	INetworkGraph to be drawn.

3.21 de.fraunhofer.iosb.trufflehog.communication.Listener< M > Interface Template Reference

Public Member Functions

- void receive (M m)

3.21.1 Detailed Description

The IListener interface works together with Notifier and INotifier to create the main communication method between threads in TruffleHog. This communication method is a variation of the observer design pattern where messages sent from the subject to the observer include a parameter of type M. The IListener is the observer in this case. It registers with the Notifier and receives messages sent from the Notifier through the receive method.

Parameters

<*M*>	The type of message to receive

3.21.2 Member Function Documentation

3.21.2.1 void de.fraunhofer.iosb.trufflehog.communication.Listener< M >.receive (M *m*)

Gets a message from a Notifier this IListener is registered to along with a parameter of type M.

Parameters

m	The message that is sent from the Notifier to the IListener.

3.22 de.fraunhofer.iosb.trufflehog.Main Class Reference

Static Public Member Functions

- static void main (String[ ] args)

3.22.1 Detailed Description

The main class in TruffleHog. It initiates everything.

3.22.2 Member Function Documentation

3.22.2.1 static void de.fraunhofer.iosb.trufflehog.Main.main (String[ ] *args*) [static]

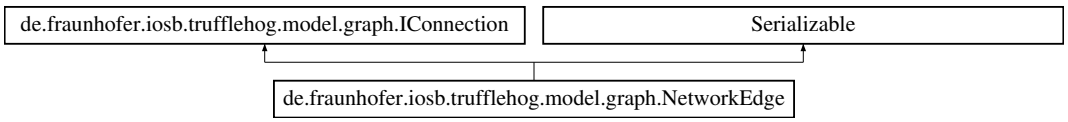
The main method of TruffleHog. It starts the program.

Parameters

args	command line arguments

3.23 de.fraunhofer.iosb.trufflehog.model.graph.NetworkEdge Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.NetworkEdge:

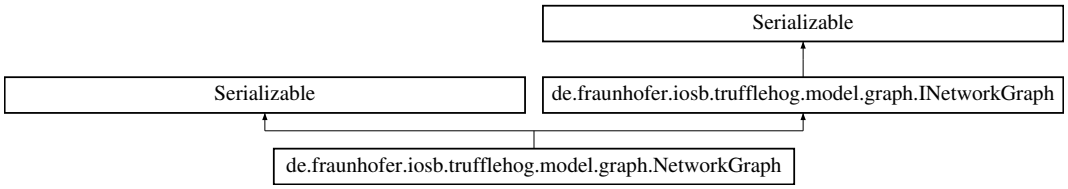


3.23.1 Detailed Description

Edge in the graph to represent a relation between two devices. Stores important statistics about the ongoing communication.

3.24 de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraph Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraph:



Public Member Functions

- void addNetworkEdge (NetworkNode from, NetworkNode to)
- void addNetworkNode (NetworkNode node)

3.24.1 Detailed Description

Stores NetworkNodes and NetworkConnections.

3.24.2 Member Function Documentation

3.24.2.1 void de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraph.addNetworkEdge (NetworkNode *from,* NetworkNode *to*)

Implements de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraph.

3.24.2.2 void de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraph.addNetworkNode (NetworkNode *node*)

Adds a new NetworkNode to the graph to represent a new device.

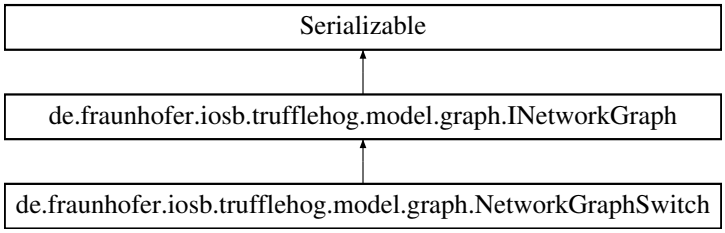
Parameters

node	NetworkNode to add p> Adds a new NetworkEdge to the graph to represent a new communication protocol used by two nodes.
from	NetworkNode sending packages
to	NetworkNode receiving packages

Implements de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraph.

3.25 de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraphSwitch Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraphSwitch:



Public Member Functions

- void viewPlayback ()
- void **viewLive** ()
- void addNetworkNode (NetworkNode node)
- void addNetworkEdge (NetworkNode from, NetworkNode to)

3.25.1 Detailed Description

Switch to jump between replay and live modes of the graph.

3.25.2 Member Function Documentation

3.25.2.1 void de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraphSwitch.addNetworkEdge (NetworkNode *from,* NetworkNode *to*)

Implements de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraph.

3.25.2.2 void de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraphSwitch.addNetworkNode (NetworkNode *node*)

Adds a new NetworkNode to the graph to represent a new device.

Parameters

node	NetworkNode to add p> Adds a new NetworkEdge to the graph to represent a new communication protocol used by two nodes.
from	NetworkNode sending packages
to	NetworkNode receiving packages

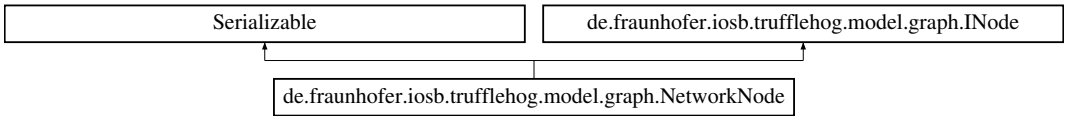
Implements de.fraunhofer.iosb.trufflehog.model.graph.INetworkGraph.

3.25.2.3 void de.fraunhofer.iosb.trufflehog.model.graph.NetworkGraphSwitch.viewPlayback ( )

p> Sets the viewing mode to live

3.26 de.fraunhofer.iosb.trufflehog.model.graph.NetworkNode Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.model.graph.NetworkNode:



Public Member Functions

- void log (Truffle truffle)

3.26.1 Detailed Description

Node in the graph to represent a device in the network. Stores important device data and logs.

3.26.2 Member Function Documentation

3.26.2.1 void de.fraunhofer.iosb.trufflehog.model.graph.NetworkNode.log (Truffle *truffle*)

Provides the internal logger to access logs and statistics.

Returns

TruffleLogger of this node

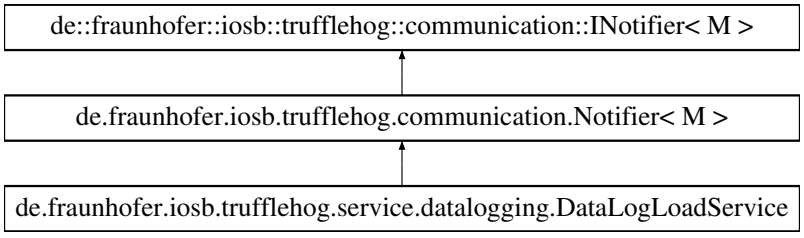
Logs a Truffle package with the internal TruffleLogger.

Parameters

truffle	Truffle to log

3.27 de.fraunhofer.iosb.trufflehog.communication.Notifier< M > Class Template Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.communication.Notifier< M >:



Public Member Functions

- void addListener (Listener listener)
- void removeListener (Listener listener)
- void notifyListeners (M message)

3.27.1 Detailed Description

The Notifier class works together with INotifier and Listener to create the main communication method between threads in TruffleHog. This communication method is a variation of the observer design pattern where messages sent from the subject to the observer include a parameter of type M. The Notifier is the subject in this case. IListeners register with the Notifier and can then be notified through the notifyListeners method. Unlike in the classic observer pattern, the notifyListeners method offers the possibility to pass along a parameter of type M.

The Notifier is the implementation of the INotifier. If a class can extend the Notifier it should.

Parameters

<*M*>	The type of message to receive.

3.27.2 Member Function Documentation

3.27.2.1 void de.fraunhofer.iosb.trufflehog.communication.Notifier< M >.addListener (Listener *listener*)

Register an IListener with this INotifier. This IListener will then be notified on the notifyListeners(M) method call.

Parameters

listener	The IListeners to register with this INotifier.

Implements de.fraunhofer.iosb.trufflehog.communication.INotifier< M >.

3.27.2.2 void de.fraunhofer.iosb.trufflehog.communication.Notifier< M >.notifyListeners (M *message*)

Notifies all IListeners that are registered with this INotifier. It sends along a message of type M

Parameters

message	The message to send along with the notification to each IListener.

Implements de.fraunhofer.iosb.trufflehog.communication.INotifier< M >.

3.27.2.3 void de.fraunhofer.iosb.trufflehog.communication.Notifier< M >.removeListener (Listener *listener*)

Removes an IListener from this INotifier. This IListener will then not be notified anymore on the notifyListeners(M) method call.

Parameters

listener	The IListeners to register with this INotifier.

Implements de.fraunhofer.iosb.trufflehog.communication.INotifier< M >.

3.28 de.fraunhofer.iosb.trufflehog.service.datalogging.SnapshotLogger Class Reference

Public Member Functions

- SnapshotLogger (GraphProxy graphProxy)

- INetworkGraph takeSnapshot ()

3.28.1 Detailed Description

The SnapshotLogger takes the current graph from the GraphProxy and takes a snapshot of it. This snapshot is then given to the DataLogger to help generate a DataLog.

3.28.2 Constructor & Destructor Documentation

3.28.2.1 de.fraunhofer.iosb.trufflehog.service.datalogging.SnapshotLogger.SnapshotLogger (GraphProxy *graphProxy*)

Creates a new SnapshotLogger object with a GraphProxy object. The GraphProxy object contains the graph (after the proxy design pattern) so that the SnapshotLogger can take a snapshot of the graph.

Parameters

graphProxy	The proxy object that contains the graph so that the SnapshotLogger can take a snapshot of it.

3.28.3 Member Function Documentation

3.28.3.1 INetworkGraph de.fraunhofer.iosb.trufflehog.service.datalogging.SnapshotLogger.takeSnapshot ( )

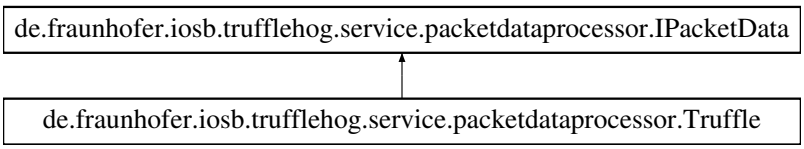
Takes a snapshot of the current graph and returns the snapshot.

Returns

A snapshot of the current graph.

3.29 de.fraunhofer.iosb.trufflehog.service.packetdataprocessor.Truffle Class Reference

Inheritance diagram for de.fraunhofer.iosb.trufflehog.service.packetdataprocessor.Truffle:



Additional Inherited Members

3.29.1 Detailed Description

This class is used to store packet data which is received from the spp_profinet snort plugin using the TruffleReceiver.

Author

Mr. X

Version

0.0

3.30 de.fraunhofer.iosb.trufflehog.service.packetdataprocessor.profinetdataprocessor.↩ TruffleReceiver Class Reference

3.30.1 Detailed Description

Created by Infinity on 26.01.2016.

Index