

[All Classes](#)

Packages

[applications](#)[core](#)[gui](#)[gui.playfield](#)[input](#)[interfaces](#)[movement](#)[movement.map](#)[report](#)[routing](#)[routing.maxprop](#)[routing.schedule](#)[ui](#)[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV NEXT

[FRAMES](#) [NO FRAMES](#)

Packages

applications	
core	Contains core classes and interfaces of the simulator.
gui	Contains the classes of Graphical User Interface.
gui.playfield	Contains the classes of Graphical User Interface's playfield - view (the graphical presentation of the nodes' locations and other information).
input	Provides interfaces and classes for reading input data from external sources.
interfaces	
movement	Contains different movement models and related classes for the simulator.
movement.map	Sub package for MapBasedMovement movement model's (and its sub classes) helper classes.
report	Contains all the report classes.
routing	Contains all the router classes who decide how to handle the messages.
routing.maxprop	Contains MaxProp routing module specific classes.
routing.schedule	
ui	Contains superclass for all user interfaces and a simple user interface(s).

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV NEXT

[FRAMES](#) [NO FRAMES](#)[All Classes](#)[ActivenessHandler](#)[ActiveRouter](#)[AdjacencyGraphvizRe](#)[Application](#)[ApplicationListener](#)[BinaryEventsReader](#)[BusControlSystem](#)[BusMovement](#)

[BusTravellerMovement](#)
[CarMovement](#)
[CBRConnection](#)
[ClusterMovement](#)
[Connection](#)
[ConnectionEvent](#)
[ConnectionListener](#)
[ConnectivityDtnsim2Repo](#)
[ConnectivityGrid](#)
[ConnectivityONEReport](#)
[ConnectivityOptimizer](#)
[ContactsDuringAnICTF](#)
[ContactsPerHourReport](#)
[ContactTimesReport](#)
[Coord](#)
[CreatedMessagesReport](#)
[Debug](#)
[DeliveredMessagesReport](#)
[DijkstraPathFinder](#)
[DirectDeliveryRouter](#)
[DistanceDelayReport](#)
[DTN2Events](#)
[DTN2Manager](#)
[DTN2Manager.EIDHost](#)
[DTN2Reporter](#)
[DTNHost](#)
[DTNSim](#)
[DTNSimGUI](#)
[DTNSimTextUI](#)
[DTNSimUI](#)
[EncountersVSUniqueEvent](#)
[EnergyAwareRouter](#)
[EnergyLevelReport](#)
[EpidemicOracleRouter](#)
[EpidemicRouter](#)
[EveningActivityControl](#)
[EveningActivityMovement](#)
[EveningTrip](#)
[EventLogControl](#)
[EventLogControlPanel](#)
[EventLogPanel](#)
[EventLogReport](#)
[EventQueue](#)
[EventQueueHandler](#)
[ExtendedMovementModel](#)
[ExternalEvent](#)
[ExternalEventsQueue](#)
[ExternalEventsReader](#)
[ExternalMovement](#)
[ExternalMovementReader](#)
[FirstContactRouter](#)
[GUIControls](#)
[HomeActivityMovement](#)
[InfoPanel](#)
[InterContactTimesReport](#)
[InterferenceLimitedInteraction](#)
[LinearFormation](#)
[MainWindow](#)
[MapBasedMovement](#)
[MapGraphic](#)
[MapNode](#)

[MapRoute](#)
[MapRouteMovement](#)
[MaxPropDijkstra](#)
[MaxPropRouter](#)
[MaxPropRouterWithEst](#)
[MeetingProbabilitySet](#)
[Message](#)
[MessageBurstGenerator](#)
[MessageCreateEvent](#)
[MessageDelayReport](#)
[MessageDeleteEvent](#)
[MessageDeliveryRepo](#)
[MessageEvent](#)
[MessageEventGenerator](#)
[MessageGraphic](#)
[MessageGraphvizReport](#)
[MessageListener](#)
[MessageLocationReport](#)
[MessageRelayEvent](#)
[MessageReport](#)
[MessageRouter](#)
[MessageStatsReport](#)
[ModuleCommunication](#)
[ModuleCommunication](#)
[MovementListener](#)
[MovementModel](#)
[MovementNs2Report](#)
[NetworkInterface](#)
[NodeChooser](#)
[NodeGraphic](#)
[OfficeActivityMovement](#)
[OneFromEachMessageG](#)
[OneToEachMessageG](#)
[ParetoRNG](#)
[PassiveRouter](#)
[Path](#)
[PathGraphic](#)
[PingApplication](#)
[PingAppReporter](#)
[PlayField](#)
[PlayFieldGraphic](#)
[PointsOfInterest](#)
[ProphetRouter](#)
[ProphetRouterWithEst](#)
[RandomWalk](#)
[RandomWaypoint](#)
[Report](#)
[RoutingInfo](#)

All Classes

[ActivenessHandler](#)
[ActiveRouter](#)
[AdjacencyGraphvizReport](#)
[Application](#)
[ApplicationListener](#)
[BinaryEventsReader](#)
[BusControlSystem](#)
[BusMovement](#)
[BusTravellerMovement](#)
[CarMovement](#)
[CBRConnection](#)
[ClusterMovement](#)
[Connection](#)
[ConnectionEvent](#)
[ConnectionListener](#)
[ConnectivityDtnsim2Report](#)
[ConnectivityGrid](#)
[ConnectivityONEReport](#)
[ConnectivityOptimizer](#)
[ContactsDuringAnICTReport](#)
[ContactsPerHourReport](#)
[ContactTimesReport](#)
[Coord](#)
[CreatedMessagesReport](#)
[Debug](#)
[DeliveredMessagesReport](#)
[DijkstraPathFinder](#)
[DirectDeliveryRouter](#)
[DistanceDelayReport](#)
[DTN2Events](#)
[DTN2Manager](#)
[DTN2Manager.EIDHost](#)
[DTN2Reporter](#)
[DTNHost](#)
[DTNSim](#)
[DTNSimGUI](#)
[DTNSimTextUI](#)
[DTNSimUI](#)
[EncountersVSUniqueEncountersReport](#)
[EnergyAwareRouter](#)
[EnergyLevelReport](#)
[EpidemicOracleRouter](#)
[EpidemicRouter](#)
[EveningActivityControlSystem](#)
[EveningActivityMovement](#)
[EveningTrip](#)
[EventLogControl](#)
[EventLogControlPanel](#)
[EventLogPanel](#)
[EventLogReport](#)
[EventQueue](#)
[EventQueueHandler](#)
[ExtendedMovementModel](#)
[ExternalEvent](#)
[ExternalEventsQueue](#)
[ExternalEventsReader](#)
[ExternalMovement](#)
[ExternalMovementReader](#)
[FirstContactRouter](#)

[GUIControls](#)
[HomeActivityMovement](#)
[InfoPanel](#)
[InterContactTimesReport](#)
[InterferenceLimitedInterface](#)
[LinearFormation](#)
[MainWindow](#)
[MapBasedMovement](#)
[MapGraphic](#)
[MapNode](#)
[MapRoute](#)
[MapRouteMovement](#)
[MaxPropDijkstra](#)
[MaxPropRouter](#)
[MaxPropRouterWithEstimation](#)
[MeetingProbabilitySet](#)
[Message](#)
[MessageBurstGenerator](#)
[MessageCreateEvent](#)
[MessageDelayReport](#)
[MessageDeleteEvent](#)
[MessageDeliveryReport](#)
[MessageEvent](#)
[MessageEventGenerator](#)
[MessageGraphic](#)
[MessageGraphvizReport](#)
[MessageListener](#)
[MessageLocationReport](#)
[MessageRelayEvent](#)
[MessageReport](#)
[MessageRouter](#)
[MessageStatsReport](#)
[ModuleCommunicationBus](#)
[ModuleCommunicationListener](#)
[MovementListener](#)
[MovementModel](#)
[MovementNs2Report](#)
[NetworkInterface](#)
[NodeChooser](#)
[NodeGraphic](#)
[OfficeActivityMovement](#)
[OneFromEachMessageGenerator](#)
[OneToEachMessageGenerator](#)
[ParetoRNG](#)
[PassiveRouter](#)
[Path](#)
[PathGraphic](#)
[PingApplication](#)
[PingAppReporter](#)
[PlayField](#)
[PlayFieldGraphic](#)
[PointsOfInterest](#)
[ProphetRouter](#)
[ProphetRouterWithEstimation](#)
[RandomWalk](#)
[RandomWaypoint](#)
[Report](#)
[RoutingInfo](#)
[RoutingInfoWindow](#)
[ScaleReferenceGraphic](#)
[ScheduleDijkstra](#)

[ScheduledUpdatesQueue](#)[ScheduleEntry](#)[ScheduleOracle](#)[Settings](#)[SettingsError](#)[ShortestPathMapBasedMovement](#)[SimClock](#)[SimError](#)[SimMap](#)[SimMenuBar](#)[SimpleBroadcastInterface](#)[SimScenario](#)[SprayAndWaitRouter](#)[StandardEventsReader](#)[StationaryMovement](#)[SwitchableMovement](#)[TotalContactTimeReport](#)[TotalEncountersReport](#)[TransportMovement](#)[Tuple](#)[UniqueEncountersReport](#)[UpdateListener](#)[VBRConnection](#)[WKTMapReader](#)[WKTRader](#)[WorkingDayMovement](#)[World](#)

[applications](#)

Classes

[PingApplication](#)

[core](#)

Interfaces

[ApplicationListener](#)[ConnectionListener](#)[MessageListener](#)[ModuleCommunicationListener](#)[MovementListener](#)[UpdateListener](#)

Classes

[Application](#)[CBRConnection](#)[Connection](#)[Coord](#)[Debug](#)[DTN2Manager](#)[DTN2Manager.EIDHost](#)[DTNHost](#)[DTNSim](#)[Message](#)[ModuleCommunicationBus](#)[NetworkInterface](#)[ParetoRNG](#)[Settings](#)[SimClock](#)[SimScenario](#)[Tuple](#)[VBRConnection](#)[World](#)

Errors

[SettingsError](#)[SimError](#)

gui[Classes](#)[DTNSimGUI](#)[EventLogControl](#)[EventLogControlPanel](#)[EventLogPanel](#)[GUIControls](#)[InfoPanel](#)[MainWindow](#)[NodeChooser](#)[RoutingInfoWindow](#)[SimMenuBar](#)

[gui.playfield](#)

Classes

[MapGraphic](#)

[MessageGraphic](#)

[NodeGraphic](#)

[PathGraphic](#)

[PlayField](#)

[PlayFieldGraphic](#)

[ScaleReferenceGraphic](#)

[input](#)

Interfaces

[EventQueue](#)[ExternalEventsReader](#)

Classes

[BinaryEventsReader](#)[ConnectionEvent](#)[DTN2Events](#)[EventQueueHandler](#)[ExternalEvent](#)[ExternalEventsQueue](#)[ExternalMovementReader](#)[MessageBurstGenerator](#)[MessageCreateEvent](#)[MessageDeleteEvent](#)[MessageEvent](#)[MessageEventGenerator](#)[MessageRelayEvent](#)[OneFromEachMessageGenerator](#)[OneToEachMessageGenerator](#)[ScheduledUpdatesQueue](#)[StandardEventsReader](#)[WKTMapReader](#)[WKTReader](#)

[interfaces](#)

Classes

[ConnectivityGrid](#)

[ConnectivityOptimizer](#)

[InterferenceLimitedInterface](#)

[SimpleBroadcastInterface](#)

[movement](#)

Interfaces

[SwitchableMovement](#)[TransportMovement](#)

Classes

[ActivenessHandler](#)[BusControlSystem](#)[BusMovement](#)[BusTravellerMovement](#)[CarMovement](#)[ClusterMovement](#)[EveningActivityControlSystem](#)[EveningActivityMovement](#)[EveningTrip](#)[ExtendedMovementModel](#)[ExternalMovement](#)[HomeActivityMovement](#)[LinearFormation](#)[MapBasedMovement](#)[MapRouteMovement](#)[MovementModel](#)[OfficeActivityMovement](#)[Path](#)[RandomWalk](#)[RandomWaypoint](#)[ShortestPathMapBasedMovement](#)[StationaryMovement](#)[WorkingDayMovement](#)

[movement.map](#)

Classes

[DijkstraPathFinder](#)
[MapNode](#)
[MapRoute](#)
[PointsOfInterest](#)
[SimMap](#)

report

Classes

[AdjacencyGraphvizReport](#)
[ConnectivityDtnsim2Report](#)
[ConnectivityONEReport](#)
[ContactsDuringAnICTReport](#)
[ContactsPerHourReport](#)
[ContactTimesReport](#)
[CreatedMessagesReport](#)
[DeliveredMessagesReport](#)
[DistanceDelayReport](#)
[DTN2Reporter](#)
[EncountersVSUniqueEncountersReport](#)
[EnergyLevelReport](#)
[EventLogReport](#)
[InterContactTimesReport](#)
[MessageDelayReport](#)
[MessageDeliveryReport](#)
[MessageGraphvizReport](#)
[MessageLocationReport](#)
[MessageReport](#)
[MessageStatsReport](#)
[MovementNs2Report](#)
[PingAppReporter](#)
[Report](#)
[TotalContactTimeReport](#)
[TotalEncountersReport](#)
[UniqueEncountersReport](#)

[routing](#)

Classes

[ActiveRouter](#)

[DirectDeliveryRouter](#)

[EnergyAwareRouter](#)

[EpidemicOracleRouter](#)

[EpidemicRouter](#)

[FirstContactRouter](#)

[MaxPropRouter](#)

[MaxPropRouterWithEstimation](#)

[MessageRouter](#)

[PassiveRouter](#)

[ProphetRouter](#)

[ProphetRouterWithEstimation](#)

[RoutingInfo](#)

[SprayAndWaitRouter](#)

[routing.maxprop](#)

Classes

[MaxPropDijkstra](#)

[MeetingProbabilitySet](#)

[routing.schedule](#)

Classes

[ScheduleDijkstra](#)

[ScheduleEntry](#)

[ScheduleOracle](#)

[ui](#)

Classes

[DTNSimTextUI](#)

[DTNSimUI](#)

movement

Class ActivenessHandler

```
java.lang.Object
└ movement.ActivenessHandler
```

```
public class ActivenessHandler
extends java.lang.Object
```

Object of this class tell the movement models when a node belonging to a certain group is active and when not.

Field Summary

<code>static java.lang.String</code>	<u>ACTIVE_TIMES_S</u>
	Active times -setting id ("activeTimes"). Syntax: start, end Multiple times can be concatenated by repeating the sequence.

Constructor Summary

[ActivenessHandler\(Settings s\)](#)

Method Summary

<code>boolean</code>	<u>isActive()</u>
	Returns true if node should be active at the moment

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

ACTIVE_TIMES_S

```
public static final java.lang.String ACTIVE_TIMES_S
```

Active times -setting id ("activeTimes").

Syntax: start, end

Multiple times can be concatenated by repeating the sequence. Time limits should be in order and should not overlap.

See Also:

[Constant Field Values](#)

Constructor Detail

ActivenessHandler

```
public ActivenessHandler(Settings s)
```

Method Detail

isActive

```
public boolean isActive()
```

Returns true if node should be active at the moment

Returns:

true if node should be active at the moment

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class ActiveRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
```

Direct Known Subclasses:

[DirectDeliveryRouter](#), [EnergyAwareRouter](#), [EpidemicOracleRouter](#), [EpidemicRouter](#), [FirstContactRouter](#), [MaxPropRouter](#), [MaxPropRouterWithEstimation](#), [ProphetRouter](#), [ProphetRouterWithEstimation](#), [SprayAndWaitRouter](#)

```
public abstract class ActiveRouter
  extends MessageRouter
```

Superclass of active routers. Contains convenience methods (e.g. [getOldestMessage\(boolean\)](#)) and watching of sending connections (see [update\(\)](#)).

Field Summary

static java.lang.String	DELETE_DELIVERED_S Delete delivered messages -setting id ("deleteDelivered").
protected boolean	deleteDelivered should messages that final recipient marks as delivered be deleted from message buffer
static java.lang.String	RESPONSE_PREFIX prefix of all response message IDs
protected java.util.ArrayList<Connection>	sendingConnections connection(s) that are currently used for sending
static int	TTL_CHECK_INTERVAL how often TTL check (discarding old messages) is performed

Fields inherited from class routing.MessageRouter

```
B_SIZE_S, DENIED_NO_SPACE, DENIED_OLD, DENIED_TTL, DENIED_UNSPECIFIED, MSG_TTL_S, msgTtl, Q_MODE_FIFO,
Q_MODE_RANDOM, RCV_OK, SEND_QUEUE_MODE_S, TRY_LATER_BUSY
```

Constructor Summary

protected	ActiveRouter(ActiveRouter r) Copy constructor.
	ActiveRouter(Settings s) Constructor.

Method Summary

protected void	addToSendingConnections(Connection con) Adds a connection to sending connections which are monitored in the update.
protected boolean	canStartTransfer() Makes rudimentary checks (that we have at least one message and one connection) about can this router start transfer.
void	changedConnection(Connection con) Called when a connection's state changes.
protected int	checkReceiving(Message m) Checks if router "wants" to start receiving message (i.e.
boolean	createNewMessage(Message m) Creates a new message to the router.
protected void	dropExpiredMessages()

		Drops messages whose TTL is less than zero.
protected Connection		exchangeDeliverableMessages() Exchanges deliverable (to final recipient) messages between this host and all hosts this host is currently connected to.
protected java.util.List<Connection>		getConnections() Returns a list of connections this host currently has with other hosts.
protected java.util.List<Tuple<Message, Connection>>		getMessagesForConnected() Returns a list of message-connections tuples of the messages whose recipient is some host that we're connected to at the moment.
protected Message		getOldestMessage(boolean excludeMsgBeingSent) Returns the oldest (by receive time) message in the message buffer (that is not being sent if excludeMsgBeingSent is true).
void		init(DTNHost host, java.util.List<MessageListener> mListeners) Initializes the router; i.e.
boolean		isSending(java.lang.String msgId) Returns true if this router is currently sending a message with msgId.
boolean		isTransferring() Returns true if this router is transferring something at the moment or some transfer has not been finalized.
protected boolean		makeRoomForMessage(int size) Removes messages from the buffer (oldest first) until there's enough space for the new message.
protected void		makeRoomForNewMessage(int size) Tries to make room for a new message.
Message		messageTransferred(java.lang.String id, DTNHost from) This method should be called (on the receiving host) after a message was successfully transferred.
int		receiveMessage(Message m, DTNHost from) Try to start receiving a message from another host.
boolean		requestDeliverableMessages(Connection con) Requests for deliverable message from this router to be sent trough a connection.
protected void		shuffleMessages(java.util.List<Message> messages) Shuffles a messages list so the messages are in random order.
protected int		startTransfer(Message m, Connection con) Tries to start a transfer of message using a connection.
protected void		transferAborted(Connection con) Method is called just before a transfer is aborted at update() due connection going down.
protected void		transferDone(Connection con) Method is called just before a transfer is finalized at update() .
protected Message		tryAllMessages(Connection con, java.util.List<Message> messages) Goes trough the messages until the other node accepts one for receiving (or doesn't accept any).
protected Connection		tryAllMessagesToAllConnections() Tries to send all messages that this router is carrying to all connections this node has.
protected Tuple<Message, Connection>		tryMessagesForConnected(java.util.List<Tuple<Message, Connection>> tuples) Tries to send messages for the connections that are mentioned in the Tuples in the order they are in the list until one of the connections starts transferring or all tuples have been tried.
protected Connection		tryMessagesToConnections(java.util.List<Message> messages, java.util.List<Connection> connections) Tries to send all given messages to all given connections.
void		update() Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.[MessageRouter](#)

```
addApplication, addToMessages, compareByQueueMode, deleteMessage, getApplications, getBufferSize,
getFreeBufferSize, getHost, getMessage, getMessageCollection, getNrofMessages, getRoutingInfo, hasMessage,
isDeliveredMessage, isIncomingMessage, messageAborted, putToIncomingBuffer, removeFromIncomingBuffer,
removeFromMessages, replicate, sendMessage, sortByQueueMode, toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Field Detail

DELETE_DELIVERED_S

```
public static final java.lang.String DELETE_DELIVERED_S
```

Delete delivered messages -setting id ("deleteDelivered"). Boolean valued. If set to true and final recipient of a message rejects it because it already has it, the message is deleted from buffer. Default=false.

See Also:

[Constant Field Values](#)

deleteDelivered

```
protected boolean deleteDelivered
```

should messages that final recipient marks as delivered be deleted from message buffer

RESPONSE_PREFIX

```
public static final java.lang.String RESPONSE_PREFIX
```

prefix of all response message IDs

See Also:

[Constant Field Values](#)

TTL_CHECK_INTERVAL

```
public static int TTL_CHECK_INTERVAL
```

how often TTL check (discarding old messages) is performed

sendingConnections

```
protected java.util.ArrayList<Connection> sendingConnections
```

connection(s) that are currently used for sending

Constructor Detail

ActiveRouter

```
public ActiveRouter(Settings s)
```

Constructor. Creates a new message router based on the settings in the given Settings object.

Parameters:

s - The settings object

ActiveRouter

```
protected ActiveRouter(ActiveRouter r)
```

Copy constructor.

Parameters:

r - The router prototype where setting values are copied from

Method Detail

init

```
public void init(DTNHost host,
                 java.util.List<MessageListener> mListeners)
```

Description copied from class: [MessageRouter](#)

Initializes the router; i.e. sets the host this router is in and message listeners that need to be informed about message related events etc.

Overrides:

[init](#) in class [MessageRouter](#)

Parameters:

host - The host this router is in

mListeners - The message listeners

changedConnection

```
public void changedConnection(Connection con)
```

Called when a connection's state changes. This version doesn't do anything but subclasses may want to override this.

Specified by:

[changedConnection](#) in class [MessageRouter](#)

Parameters:

con - The connection that changed

requestDeliverableMessages

```
public boolean requestDeliverableMessages(Connection con)
```

Description copied from class: [MessageRouter](#)

Requests for deliverable message from this router to be sent trough a connection.

Overrides:

[requestDeliverableMessages](#) in class [MessageRouter](#)

Parameters:

con - The connection to send the messages trough

Returns:

True if this router started a transfer, false if not

createNewMessage

```
public boolean createNewMessage(Message m)
```

Description copied from class: [MessageRouter](#)

Creates a new message to the router.

Overrides:

[createNewMessage](#) in class [MessageRouter](#)

Parameters:

m - The message to create

Returns:

True if the creation succeeded, false if not (e.g. the message was too big for the buffer)

receiveMessage

```
public int receiveMessage(Message m,
                         DTNHost from)
```

Description copied from class: [MessageRouter](#)

Try to start receiving a message from another host.

Overrides:

[receiveMessage](#) in class [MessageRouter](#)

Parameters:

m - Message to put in the receiving buffer
from - Who the message is from

Returns:

Value zero if the node accepted the message (RCV_OK), value less than zero if node rejected the message (e.g. DENIED_OLD), value bigger than zero if the other node should try later (e.g. TRY_LATER_BUSY).

messageTransferred

```
public Message messageTransferred(java.lang.String id,
                                   DTNHost from)
```

Description copied from class: [MessageRouter](#)

This method should be called (on the receiving host) after a message was successfully transferred. The transferred message is put to the message buffer unless this host is the final recipient of the message.

Overrides:

[messageTransferred](#) in class [MessageRouter](#)

Parameters:

id - Id of the transferred message
from - Host the message was from (previous hop)

Returns:

The message that this host received

getConnections

```
protected java.util.List<Connection> getConnections()
```

Returns a list of connections this host currently has with other hosts.

Returns:

a list of connections this host currently has with other hosts

startTransfer

```
protected int startTransfer(Message m,
                           Connection con)
```

Tries to start a transfer of message using a connection. If starting succeeds, the connection is added to the watch list of active connections

Parameters:

m - The message to transfer
con - The connection to use

Returns:

the value returned by [Connection.startTransfer\(DTNHost, Message\)](#)

canStartTransfer

```
protected boolean canStartTransfer()
```

Makes rudimentary checks (that we have at least one message and one connection) about can this router start transfer.

Returns:

True if router can start transfer, false if not

checkReceiving

```
protected int checkReceiving(Message m)
```

Checks if router "wants" to start receiving message (i.e. router isn't transferring, doesn't have the message and has room for it).

Parameters:

m - The message to check

Returns:

A return code similar to `MessageRouter.receiveMessage(Message, DTNHost)`, i.e. `MessageRouter.RCV_OK` if receiving seems to be OK, `TRY_LATER_BUSY` if router is transferring, `DENIED_OLD` if the router is already carrying the message or it has been delivered to this router (as final recipient), or `DENIED_NO_SPACE` if the message does not fit into buffer

makeRoomForMessage

```
protected boolean makeRoomForMessage(int size)
```

Removes messages from the buffer (oldest first) until there's enough space for the new message.

Parameters:

size - Size of the new message transferred, the transfer is aborted before message is removed

Returns:

True if enough space could be freed, false if not

dropExpiredMessages

```
protected void dropExpiredMessages()
```

Drops messages whose TTL is less than zero.

makeRoomForNewMessage

```
protected void makeRoomForNewMessage(int size)
```

Tries to make room for a new message. Current implementation simply calls `makeRoomForMessage(int)` and ignores the return value. Therefore, if the message can't fit into buffer, the buffer is only cleared from messages that are not being sent.

Parameters:

size - Size of the new message

getOldestMessage

```
protected Message getOldestMessage(boolean excludeMsgBeingSent)
```

Returns the oldest (by receive time) message in the message buffer (that is not being sent if excludeMsgBeingSent is true).

Parameters:

excludeMsgBeingSent - If true, excludes message(s) that are being sent from the oldest message check (i.e. if oldest message is being sent, the second oldest message is returned)

Returns:

The oldest message or null if no message could be returned (no messages in buffer or all messages in buffer are being sent and excludeMsgBeingSent is true)

getMessagesForConnected

```
protected java.util.List<Tuple<Message, Connection>> getMessagesForConnected()
```

Returns a list of message-connections tuples of the messages whose recipient is some host that we're connected to at the moment.

Returns:

a list of message-connections tuples

tryMessagesForConnected

```
protected Tuple<Message, Connection> tryMessagesForConnected(java.util.List<Tuple<Message, Connection>> tuples)
```

Tries to send messages for the connections that are mentioned in the Tuples in the order they are in the list until one of the connections starts transferring or all tuples have been tried.

Parameters:

tuples - The tuples to try

Returns:

The tuple whose connection accepted the message or null if none of the connections accepted the message that was meant for them.

tryAllMessages

```
protected Message tryAllMessages(Connection con,
                                java.util.List<Message> messages)
```

Goes through the messages until the other node accepts one for receiving (or doesn't accept any). If a transfer is started, the connection is included in the list of sending connections.

Parameters:

con - Connection through which the messages are sent

messages - A list of messages to try

Returns:

The message whose transfer was started or null if no transfer was started.

tryMessagesToConnections

```
protected Connection tryMessagesToConnections(java.util.List<Message> messages,
                                              java.util.List<Connection> connections)
```

Tries to send all given messages to all given connections. Connections are first iterated in the order they are in the list and for every connection, the messages are tried in the order they are in the list. Once an accepting connection is found, no other connections or messages are tried.

Parameters:

messages - The list of Messages to try

connections - The list of Connections to try

Returns:

The connections that started a transfer or null if no connection accepted a message.

tryAllMessagesToAllConnections

```
protected Connection tryAllMessagesToAllConnections()
```

Tries to send all messages that this router is carrying to all connections this node has. Messages are ordered using the [MessageRouter.sortByQueueMode\(List\)](#). See [tryMessagesToConnections\(List, List\)](#) for sending details.

Returns:

The connections that started a transfer or null if no connection accepted a message.

exchangeDeliverableMessages

```
protected Connection exchangeDeliverableMessages()
```

Exchanges deliverable (to final recipient) messages between this host and all hosts this host is currently connected to. First all messages from this host are checked and then all other hosts are asked for messages to this host. If a transfer is started, the search ends.

Returns:

A connection that started a transfer or null if no transfer was started

shuffleMessages

```
protected void shuffleMessages(java.util.List<Message> messages)
```

Shuffles a messages list so the messages are in random order.

Parameters:

messages - The list to sort and shuffle

addToSendingConnections

```
protected void addToSendingConnections(Connection con)
```

Adds a connections to sending connections which are monitored in the update.

Parameters:

con - The connection to add

See Also:

[update\(\)](#)

isTransferring

```
public boolean isTransferring()
```

Returns true if this router is transferring something at the moment or some transfer has not been finalized.

Returns:

true if this router is transferring something

isSending

```
public boolean isSending(java.lang.String msgId)
```

Returns true if this router is currently sending a message with msgId.

Parameters:

msgId - The ID of the message

Returns:

True if the message is being sent false if not

update

```
public void update()
```

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [MessageRouter](#)

See Also:

[addToSendingConnections\(Connection\)](#)

transferAborted

```
protected void transferAborted(Connection con)
```

Method is called just before a transfer is aborted at [update\(\)](#) due connection going down. This happens on the sending host. Subclasses that are interested of the event may want to override this.

Parameters:

con - The connection whose transfer was aborted

transferDone

```
protected void transferDone(Connection con)
```

Method is called just before a transfer is finalized at [update\(\)](#). Subclasses that are interested of the event may want to override this.

Parameters:

con - The connection whose transfer was finalized

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | CONSTR | METHOD

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class AdjacencyGraphvizReport

```
java.lang.Object
└ report.Report
    └ report.AdjacencyGraphvizReport
```

All Implemented Interfaces:

[ConnectionListener](#)

```
public class AdjacencyGraphvizReport
extends Report
implements ConnectionListener
```

Generates Graphviz compatible graph from connections. Connections that happen during the warm up period are ignored.

Field Summary

static java.lang.String	GRAPH_NAME
Name of the graphviz report ("adjgraph")	

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[AdjacencyGraphvizReport\(\)](#)

Constructor.

Method Summary

void	done()	Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
void	hostsConnected(DTNHost host1, DTNHost host2)	Method is called when two hosts are connected.
void	hostsDisconnected(DTNHost host1, DTNHost host2)	Method is called when connection between hosts is disconnected.
protected void	init()	Initializes the report output.
void	setAllHosts(java.util.Collection<DTNHost> hosts)	Sets all hosts that should be in the graph at least once

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,  

getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,  

setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

Field Detail

GRAPH_NAME

```
public static final java.lang.String GRAPH_NAME
```

Name of the graphviz report ("adjgraph")

See Also:

[Constant Field Values](#)

Constructor Detail

AdjacencyGraphvizReport

```
public AdjacencyGraphvizReport()
```

Constructor.

Method Detail

init

```
protected void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

hostsConnected

```
public void hostsConnected(DTNHost host1,  

DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the connection

host2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost host1,
                            DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

setAllHosts

```
public void setAllHosts(java.util.Collection<DTNHost> hosts)
```

Sets all hosts that should be in the graph at least once

Parameters:

hosts - Collection of hosts

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Class Application

```
java.lang.Object
└ core.Application
```

Direct Known Subclasses:

[PingApplication](#)

```
public abstract class Application
extends java.lang.Object
```

Base class for applications. Nodes that have an application running will forward all incoming messages to the application `handle()` method before they are processed further. The application can change the properties of the message before returning it or return null to signal to the router that it wants the message to be dropped.

In addition, the application's `update()` method is called every simulation cycle.

Configuration of application is done by picking a unique application instance name (e.g., `mySimpleApp`) and setting its `type` property to the concrete application class: `mySimpleApp.type = SimpleApplication`. These application instances can be assigned to node groups using the `Group.application` setting: `Group1.application = mySimpleApp`.

Field Summary

java.lang.String	appID
------------------	-----------------------

Constructor Summary

[Application\(\)](#)

[Application\(Application app\)](#)

Copy constructor.

Method Summary

java.lang.String	getAppID()
------------------	----------------------------

Returns an unique application ID.

java.util.List< ApplicationListener >	getAppListeners()
---	-----------------------------------

abstract Message	handle(Message msg, DTNHost host)
----------------------------------	---

This method handles application functionality related to processing of the bundle.

abstract Application	replicate()
--------------------------------------	-----------------------------

void	sendEventToListeners(java.lang.String event, java.lang.Object params, DTNHost host)
------	---

	Sends an event to all listeners.
void	setAppID (java.lang.String appID) Sets the application ID.
void	setAppListeners (java.util.List< ApplicationListener > aListeners)
abstract void	update (DTNHost host) Called every simulation cycle.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Field Detail**appID**public java.lang.String **appID****Constructor Detail****Application**public **Application()****Application**public **Application**([Application](#) app)

Copy constructor.

Parameters:

app -

Method Detail**handle**public abstract [Message](#) **handle**([Message](#) msg,
[DTNHost](#) host)

This method handles application functionality related to processing of the bundle. Application handles a messages, which arrives to the node hosting this application. After performing application specific handling, this method returns a list of messages. If node wishes to continue forwarding the incoming

Parameters:

msg - The incoming message.

host - The host this application instance is attached to.

Returns:

the (possibly modified) message to forward or `null` if the application wants the router to stop forwarding the message.

update

```
public abstract void update(DTNHost host)
```

Called every simulation cycle.

Parameters:

host - The host this application instance is attached to.

getAppID

```
public java.lang.String getAppID\(\)
```

Returns an unique application ID. The application will only receive messages with this application ID. If the AppID is set to null the application will receive all messages.

Returns:

Application ID.

setAppID

```
public void setAppID\(java.lang.String appID\)
```

Sets the application ID. Should only set once when the application is created. Changing the value during simulation runtime is not recommended unless you really know what you're doing.

Parameters:

appID -

replicate

```
public abstract Application replicate\(\)
```

setAppListeners

```
public void setAppListeners\(java.util.List<ApplicationListener> aListeners\)
```

getAppListeners

```
public java.util.List<ApplicationListener> getAppListeners\(\)
```

sendEventToListeners

```
public void sendEventToListeners\(java.lang.String event,
                               java.lang.Object params,
                               DTNHost host\)
```

Sends an event to all listeners.

Parameters:

event - The event to send.

params - Any additional parameters to send.

host - The host which where the app is running.

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Interface ApplicationListener

All Known Implementing Classes:

[PingAppReporter](#)

```
public interface ApplicationListener
```

Interface for classes that want to be informed about messages between hosts.

Report classes wishing to receive application events should implement this interface. Note that the application event names are defined by the applications so any class wishing to interpret them must know the application.

Method Summary

void	gotEvent (java.lang.String event, java.lang.Object params, Application app, DTNHost host)
	Application has generated an event.

Method Detail

gotEvent

```
void gotEvent(java.lang.String event,
              java.lang.Object params,
              Application app,
              DTNHost host)
```

Application has generated an event.

Parameters:

- event - Event name.
- params - Additional parameters for the event
- app - Application instance that generated the event.
- host - The host this application instance is running on.

input

Class BinaryEventsReader

```
java.lang.Object
└─ input.BinaryEventsReader
```

All Implemented Interfaces:

[ExternalEventsReader](#)

```
public class BinaryEventsReader
extends java.lang.Object
implements ExternalEventsReader
```

Reads External Events from a binary file. Can also create binary files from a list of external events.

Field Summary

static java.lang.String	BINARY_EXT
	Extension of binary external events file

Constructor Summary

[BinaryEventsReader](#)(java.io.File eventsFile)

Constructor.

Method Summary

void	close() Closes the input file streams of the reader.
static boolean	isBinaryEeFile (java.io.File file) Checks if the given file is a binary external events file
java.util.List< ExternalEvent >	readEvents (int nrof) Read events from a binary file created with storeBinaryFile method
static void	storeToBinaryFile (java.lang.String fileName, java.util.List< ExternalEvent > events) Stores the events to a binary file

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Field Detail

BINARY_EXT

```
public static final java.lang.String BINARY_EXT
```

Extension of binary external events file

See Also:

[Constant Field Values](#)

Constructor Detail

BinaryEventsReader

```
public BinaryEventsReader(java.io.File eventsFile)
```

Constructor.

Parameters:

eventsFile - The file where the events are read

Method Detail

readEvents

```
public java.util.List<ExternalEvent> readEvents(int nrof)
```

Read events from a binary file created with storeBinaryFile method

Specified by:

[readEvents](#) in interface [ExternalEventsReader](#)

Parameters:

nrof - Maximum number of events to read

Returns:

Events in an ArrayList (empty list if didn't read any)

See Also:

[storeToBinaryFile\(String, List\)](#)

isBinaryEeFile

```
public static boolean isBinaryEeFile(java.io.File file)
```

Checks if the given file is a binary external events file

Parameters:

file - The file to check

Returns:

True if the file is a binary ee file, false if not

storeToBinaryFile

```
public static void storeToBinaryFile(java.lang.String fileName,
                                     java.util.List<ExternalEvent> events)
                                     throws java.io.IOException
```

Stores the events to a binary file

Parameters:

fileName - Path to the file where the events are stored

events - List of events to store

Throws:

java.io.IOException - if something in storing went wrong

close

```
public void close()
```

Description copied from interface: [ExternalEventsReader](#)

Closes the input file streams of the reader.

Specified by:

[close](#) in interface [ExternalEventsReader](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | CONSTR | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | CONSTR | [METHOD](#)

movement

Class BusControlSystem

```
java.lang.Object
└ movement.BusControlSystem
```

```
public class BusControlSystem
extends java.lang.Object
```

This class controls busses and passengers that can use the bus. There can be many bus BusControlSystems, but a bus or passenger can only belong to one system.

Field Summary

static java.lang.String	BUS CONTROL SYSTEM NR
-------------------------	---------------------------------------

Method Summary

void	busHasStopped (int busID, Coord busStop, Path nextPath) Called by busses belonging to this system every time the bus has stopped.
static BusControlSystem	getBusControlSystem (int systemID) Returns a reference to a BusControlSystem with ID provided as parameter.
java.util.List< Coord >	getBusStops ()
SimMap	getMap () Get the underlying map of the system
void	registerBus (BusMovement bus) Registers a bus to be part of a bus control system
void	registerTraveller (BusTravellerMovement traveller) Registers a traveller/passenger to be part of a bus control system
static void	reset ()
void	setBusStops (java.util.List< Coord > busStops) Set the bus stops that belong to this system
void	setMap (SimMap map) Provide the system with the map

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

BUS_CONTROL_SYSTEM_NR

```
public static final java.lang.String BUS_CONTROL_SYSTEM_NR
```

See Also:

[Constant Field Values](#)

Method Detail

reset

```
public static void reset()
```

busHasStopped

```
public void busHasStopped(int busID,
                          Coord busStop,
                          Path nextPath)
```

Called by busses belonging to this system every time the bus has stopped. It calls every passengers enterBus() method so that the passengers can enter the bus if they want to.

Parameters:

- busID - Unique identifier of the bus
- busStop - Coordinates of the bus stop
- nextPath - The path to the next stop

getBusControlSystem

```
public static BusControlSystem getBusControlSystem(int systemID)
```

Returns a reference to a BusControlSystem with ID provided as parameter. If a system does not already exist with the requested ID, a new one is created.

Parameters:

- systemID - unique ID of the system

Returns:

The bus control system with the provided ID

registerBus

```
public void registerBus(BusMovement bus)
```

Registers a bus to be part of a bus control system

Parameters:

- bus - The bus to register

registerTraveller

```
public void registerTraveller(BusTravellerMovement traveller)
```

Registers a traveller/passenger to be part of a bus control system

Parameters:

traveller - The traveller to register

setMap

```
public void setMap(SimMap map)
```

Provide the system with the map

Parameters:

map -

getMap

```
public SimMap getMap()
```

Get the underlying map of the system

Returns:

The map

getBusStops

```
public java.util.List<Coord> getBusStops()
```

Returns:

A list of all bus stops belonging to this system

setBusStops

```
public void setBusStops(java.util.List<Coord> busStops)
```

Set the bus stops that belong to this system

Parameters:

busStops -

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | CONSTR | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | CONSTR | [METHOD](#)

movement

Class BusMovement

```
java.lang.Object
  └── movement.MovementModel
    └── movement.MapBasedMovement
      └── movement.MapRouteMovement
        └── movement.BusMovement
```

All Implemented Interfaces:

[SwitchableMovement](#)

```
public class BusMovement
extends MapRouteMovement
```

This class controls the movement of busses. It informs the bus control system the bus is registered with every time the bus stops.

Field Summary

Fields inherited from class movement.MapRouteMovement

[ROUTE_FILE_S](#), [ROUTE_FIRST_STOP_S](#), [ROUTE_TYPE_S](#)

Fields inherited from class movement.MapBasedMovement

[backAllowed](#), [FILE_S](#), [lastMapNode](#), [MAP_BASE_MOVEMENT_NS](#), [MAP_SELECT_S](#), [maxPathLength](#), [minPathLength](#), [NROF_FILES_S](#)

Fields inherited from class movement.MovementModel

[comBus](#), [DEF_SPEEDS](#), [DEF_WAIT_TIMES](#), [maxSpeed](#), [maxWaitTime](#), [minSpeed](#), [minWaitTime](#), [MOVEMENT_MODEL_NS](#), [rng](#), [RNG_SEED](#), [SPEED](#), [WAIT_TIME](#), [WORLD_SIZE](#)

Constructor Summary

[BusMovement\(BusMovement proto\)](#)

Create a new instance from a prototype

[BusMovement\(Settings settings\)](#)

Creates a new instance of BusMovement

Method Summary

int getID()	Returns unique ID of the bus
Coord getInitialLocation()	Returns the first stop on the route
Path getPath()	Returns a new path by this movement model or null if no new path could be constructed at

the moment (node should wait where it is).

[BusMovement](#)

[replicate\(\)](#)

Creates a replicate of the movement model.

Methods inherited from class movement.[MapRouteMovement](#)

[getLastLocation](#), [getStops](#)

Methods inherited from class movement.[MapBasedMovement](#)

[getMap](#), [getOkMapNodeTypes](#), [isReady](#), [selectRandomOkNode](#), [setLocation](#)

Methods inherited from class movement.[MovementModel](#)

[generateSpeed](#), [generateWaitTime](#), [getComBus](#), [getMaxX](#), [getMaxY](#), [isActive](#), [nextPathAvailable](#), [reset](#), [setComBus](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface movement.[SwitchableMovement](#)

[isReady](#), [setLocation](#)

Constructor Detail

BusMovement

```
public BusMovement(Settings settings)
```

Creates a new instance of BusMovement

Parameters:

settings -

BusMovement

```
public BusMovement(BusMovement proto)
```

Create a new instance from a prototype

Parameters:

proto -

Method Detail

getInitialLocation

```
public Coord getInitialLocation()
```

Description copied from class: [MapRouteMovement](#)

Returns the first stop on the route

Overrides:

[getInitialLocation](#) in class [MapRouteMovement](#)

Returns:

The initial coordinates for a node

getPath

public [Path](#) [getPath\(\)](#)

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Overrides:

[getPath](#) in class [MapRouteMovement](#)

Returns:

A new path or null

replicate

public [BusMovement](#) [replicate\(\)](#)

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Overrides:

[replicate](#) in class [MapRouteMovement](#)

Returns:

A new movement model with the same settings as this model

getID

public int [getID\(\)](#)

Returns unique ID of the bus

Returns:

unique ID of the bus

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

Hierarchy For All Packages

Package Hierarchies:

[applications](#), [core](#), [gui](#), [gui.playfield](#), [input](#), [interfaces](#), [movement](#), [movement.map](#), [report](#), [routing](#), [routing.maxprop](#), [routing.schedule](#), [ui](#)

Class Hierarchy

- [java.lang.Object](#)
 - [movement.ActiveHandler](#)
 - [core.Application](#)
 - [applications.PingApplication](#)
 - [input.BinaryEventsReader](#) (implements [input.ExternalEventsReader](#))
 - [movement.BusControlSystem](#)
 - [java.awt.Component](#) (implements [java.awt.image.ImageObserver](#), [java.awt.MenuContainer](#), [java.io.Serializable](#))
 - [java.awt.Container](#)
 - [javax.swing.JComponent](#) (implements [java.io.Serializable](#))
 - [javax.swing.JMenuBar](#) (implements [javax.accessibility.Accessible](#), [javax.swing.MenuElement](#))
 - [gui.SimMenuBar](#) (implements [java.awt.event.ActionListener](#))
 - [javax.swing.JPanel](#) (implements [javax.accessibility.Accessible](#))
 - [gui.EventLogControlPanel](#) (implements [java.awt.event.ActionListener](#))
 - [gui.EventLogPanel](#) (implements [java.awt.event.ActionListener](#), [core.ConnectionListener](#), [core.MessageListener](#))
 - [gui.GUIControls](#) (implements [java.awt.event.ActionListener](#), [javax.swing.event.ChangeListener](#))
 - [gui.InfoPanel](#) (implements [java.awt.event.ActionListener](#))
 - [gui.NodeChooser](#) (implements [java.awt.event.ActionListener](#))
 - [gui.playfield.PlayField](#)
 - [java.awt.Window](#) (implements [javax.accessibility.Accessible](#))
 - [java.awt.Frame](#) (implements [java.awt.MenuContainer](#))
 - [javax.swing.JFrame](#) (implements [javax.accessibility.Accessible](#), [javax.swing.RootPaneContainer](#), [javax.swing.WindowConstants](#))
 - [gui.MainWindow](#)
 - [gui.RoutingInfoWindow](#) (implements [java.awt.event.ActionListener](#))
 - [core.Connection](#)
 - [core.CBRCConnection](#)
 - [core.VBRCConnection](#)
 - [interfaces.ConnectivityGrid.GridCell](#)
 - [interfaces.ConnectivityOptimizer](#)
 - [interfaces.ConnectivityGrid](#)
 - [report.ContactTimesReport.ConnectionInfo](#)
 - [core.Coord](#) (implements [java.lang.Cloneable](#), [java.lang.Comparable<T>](#))
 - [core.Debug](#)
 - [movement.map.DijkstraPathFinder](#)
 - [input.DTN2Events](#) (implements [input.EventQueue](#))
 - [input.DTN2Events.ParserHandler](#)
 - [core.DTN2Manager](#)
 - [core.DTN2Manager.EIDHost](#)

- core.[DTNHost](#) (implements java.lang.Comparable<T>)
- core.[DTNSim](#)
- ui.[DTNSimUI](#)
 - gui.[DTNSimGUI](#)
 - ui.[DTNSimTextUI](#)
- movement.[EveningActivityControlSystem](#)
- movement.[EveningTrip](#)
- gui.[EventLogControl](#)
- input.[EventQueueHandler](#)
- input.[ExternalEvent](#) (implements java.lang.Comparable<T>, java.io.Serializable)
 - input.[ConnectionEvent](#)
 - input.[MessageEvent](#)
 - input.[MessageCreateEvent](#)
 - input.[MessageDeleteEvent](#)
 - input.[MessageRelayEvent](#)
- input.[ExternalEventsQueue](#) (implements input.[EventQueue](#))
- input.[ExternalMovementReader](#)
- movement.map.[MapNode](#) (implements java.lang.Comparable<T>)
- movement.map.[MapRoute](#)
- routing.maxprop.[MaxPropDijkstra](#)
- routing.maxprop.[MeetingProbabilitySet](#)
- core.[Message](#) (implements java.lang.Comparable<T>)
- input.[MessageEventGenerator](#) (implements input.[EventQueue](#))
 - input.[MessageBurstGenerator](#)
 - input.[OneFromEachMessageGenerator](#)
 - input.[OneToEachMessageGenerator](#)
- routing.[MessageRouter](#)
 - routing.[ActiveRouter](#)
 - routing.[DirectDeliveryRouter](#)
 - routing.[EnergyAwareRouter](#) (implements core.[ModuleCommunicationListener](#))
 - routing.[EpidemicOracleRouter](#)
 - routing.[EpidemicRouter](#)
 - routing.[FirstContactRouter](#)
 - routing.[MaxPropRouter](#)
 - routing.[MaxPropRouterWithEstimation](#)
 - routing.[ProphetRouter](#)
 - routing.[ProphetRouterWithEstimation](#)
 - routing.[SprayAndWaitRouter](#)
 - routing.[PassiveRouter](#)
- core.[ModuleCommunicationBus](#)
- movement.[MovementModel](#)
 - movement.[ExtendedMovementModel](#)
 - movement.[WorkingDayMovement](#)
 - movement.[ExternalMovement](#)
 - movement.[LinearFormation](#)
 - movement.[MapBasedMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[BusTravellerMovement](#) (implements movement.[SwitchableMovement](#), movement.[TransportMovement](#))
 - movement.[CarMovement](#) (implements movement.[SwitchableMovement](#), movement.[TransportMovement](#))
 - movement.[EveningActivityMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[HomeActivityMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[MapRouteMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[BusMovement](#)
 - movement.[OfficeActivityMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[ShortestPathMapBasedMovement](#) (implements movement.[SwitchableMovement](#))

- movement.[RandomWalk](#) (implements movement.[SwitchableMovement](#))
 - movement.[RandomWaypoint](#)
 - movement.[ClusterMovement](#)
 - movement.[StationaryMovement](#)
- core.[NetworkInterface](#) (implements core.[ModuleCommunicationListener](#))
 - interfaces.[InterferenceLimitedInterface](#)
 - interfaces.[SimpleBroadcastInterface](#)
- core.[ParetoRNG](#)
- movement.[Path](#)
- gui.playfield.[PlayFieldGraphic](#)
 - gui.playfield.[MapGraphic](#)
 - gui.playfield.[MessageGraphic](#)
 - gui.playfield.[NodeGraphic](#)
 - gui.playfield.[PathGraphic](#)
 - gui.playfield.[ScaleReferenceGraphic](#)
- movement.map.[PointsOfInterest](#)
- report.[Report](#)
 - report.[AdjacencyGraphvizReport](#) (implements core.[ConnectionListener](#))
 - report.[ConnectivityDtnsim2Report](#) (implements core.[ConnectionListener](#))
 - report.[ConnectivityONEReport](#) (implements core.[ConnectionListener](#))
 - report.[ContactsDuringAnICTReport](#) (implements core.[ConnectionListener](#), core.[UpdateListener](#))
 - report.[ContactsPerHourReport](#) (implements core.[ConnectionListener](#))
 - report.[ContactTimesReport](#) (implements core.[ConnectionListener](#))
 - report.[InterContactTimesReport](#)
 - report.[TotalContactTimeReport](#) (implements core.[UpdateListener](#))
 - report.[CreatedMessagesReport](#) (implements core.[MessageListener](#))
 - report.[DeliveredMessagesReport](#) (implements core.[MessageListener](#))
 - report.[DistanceDelayReport](#) (implements core.[MessageListener](#))
 - report.[DTN2Reporter](#) (implements core.[MessageListener](#))
 - report.[EncountersVSUniqueEncountersReport](#) (implements core.[ConnectionListener](#), core.[UpdateListener](#))
 - report.[EnergyLevelReport](#) (implements core.[UpdateListener](#))
 - report.[EventLogReport](#) (implements core.[ConnectionListener](#), core.[MessageListener](#))
 - report.[MessageDelayReport](#) (implements core.[MessageListener](#))
 - report.[MessageDeliveryReport](#) (implements core.[MessageListener](#))
 - report.[MessageGraphvizReport](#) (implements core.[MessageListener](#))
 - report.[MessageLocationReport](#) (implements core.[UpdateListener](#))
 - report.[MessageReport](#) (implements core.[MessageListener](#))
 - report.[MessageStatsReport](#) (implements core.[MessageListener](#))
 - report.[MovementNs2Report](#) (implements core.[MovementListener](#))
 - report.[PingAppReporter](#) (implements core.[ApplicationListener](#))
 - report.[TotalEncountersReport](#) (implements core.[ConnectionListener](#), core.[UpdateListener](#))
 - report.[UniqueEncountersReport](#) (implements core.[ConnectionListener](#), core.[UpdateListener](#))
- routing.[RoutingInfo](#)
- routing.schedule.[ScheduleDijkstra](#)
- input.[ScheduledUpdatesQueue](#) (implements input.[EventQueue](#))
- routing.schedule.[ScheduleEntry](#) (implements java.io.Serializable)
- routing.schedule.[ScheduleOracle](#) (implements java.io.Serializable)
- core.[Settings](#)
- core.[SimClock](#)
- movement.map.[SimMap](#) (implements java.io.Serializable)
- core.[SimScenario](#) (implements java.io.Serializable)
- input.[StandardEventsReader](#) (implements input.[ExternalEventsReader](#))
- java.lang.Throwable (implements java.io.Serializable)
 - java.lang.Error
 - java.lang.AssertionError

- core.[SimError](#)
 - core.[SettingsError](#)
- core.[Tuple](#)<K,V>
- input.[WKTReader](#)
 - input.[WKTMapReader](#)
- core.[World](#)

Interface Hierarchy

- core.[ApplicationListener](#)
- core.[ConnectionListener](#)
- input.[EventQueue](#)
- input.[ExternalEventsReader](#)
- core.[MessageListener](#)
- core.[ModuleCommunicationListener](#)
- core.[MovementListener](#)
- movement.[SwitchableMovement](#)
 - movement.[TransportMovement](#)
- core.[UpdateListener](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV NEXT

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV NEXT

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Deprecated API

Contents

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV NEXT

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

A

[**ABORT**](#) - Static variable in class input.[StandardEventsReader](#)

Identifier of message transfer aborted event ("A")

[**ABORTED**](#) - Static variable in class input.[MessageRelayEvent](#)

Message relay stage constant for aborted delivery

[**abortTransfer\(\)**](#) - Method in class core.[CBRConnection](#)

Aborts the transfer of the currently transferred message.

[**abortTransfer\(\)**](#) - Method in class core.[Connection](#)

Aborts the transfer of the currently transferred message.

[**ABOUT_TEXT**](#) - Static variable in class gui.[SimMenuBar](#)

GPLv3 license text for about window

[**ABOUT_TITLE**](#) - Static variable in class gui.[SimMenuBar](#)

title of the about window

[**actionPerformed\(ActionEvent\)**](#) - Method in class gui.[EventLogControlPanel](#)

[**actionPerformed\(ActionEvent\)**](#) - Method in class gui.[EventLogPanel](#)

Action listener for log entry (host & message) buttons

[**actionPerformed\(ActionEvent\)**](#) - Method in class gui.[GUIControls](#)

[**actionPerformed\(ActionEvent\)**](#) - Method in class gui.[InfoPanel](#)

[**actionPerformed\(ActionEvent\)**](#) - Method in class gui.[NodeChooser](#)

Action listener method for buttons and node set chooser

[**actionPerformed\(ActionEvent\)**](#) - Method in class gui.[RoutingInfoWindow](#)

[**actionPerformed\(ActionEvent\)**](#) - Method in class gui.[SimMenuBar](#)

[**ACTIVE_TIMES_S**](#) - Static variable in class movement.[ActivenessHandler](#)

Active times -setting id ("activeTimes").

Syntax: start, end

Multiple times can be concatenated by repeating the sequence.

[**ActivenessHandler**](#) - Class in [movement](#)

Object of this class tell the movement models when a node belonging to a certain group is active and when not.

[**ActivenessHandler\(Settings\)**](#) - Constructor for class movement.[ActivenessHandler](#)

[**ActiveRouter**](#) - Class in [routing](#)

Superclass of active routers.

[**ActiveRouter\(Settings\)**](#) - Constructor for class routing.[ActiveRouter](#)

Constructor.

[**ActiveRouter\(ActiveRouter\)**](#) - Constructor for class routing.[ActiveRouter](#)

Copy constructor.

[**addApplication\(Application\)**](#) - Method in class routing.[MessageRouter](#)

Adds an application to the attached applications list.

[**addApplicationListener\(ApplicationListener\)**](#) - Method in class core.[SimScenario](#)

Adds a new application event listener for all nodes.

[**addBundle\(String, Bundle\)**](#) - Static method in class core.[DTN2Manager](#)

Stores a reference to a bundle corresponding to the given message.

[**addConnection\(DTNHost, DTNHost\)**](#) - Method in class report.[ContactTimesReport](#)

[**addConnectionListener\(ConnectionListener\)**](#) - Method in class core.[SimScenario](#)

Adds a new connection listener for all nodes

[**addControl\(String, boolean, boolean\)**](#) - Method in class gui.[EventLogControlPanel](#)

Adds a new filter&pause control

[**addControl\(String\)**](#) - Method in class gui.[EventLogControlPanel](#)

Adds a new filter&pause control with initially "show" checked but "pause" unchecked

[**addEntry\(double, int, int, int, double\)**](#) - Method in class routing.schedule.[ScheduleOracle](#)

Adds a new schedule entry to the oracle

[**addEntry\(double, int, int, double\)**](#) - Method in class routing.schedule.[ScheduleOracle](#)

Adds a new schedule entry to the oracle

[**addEveningActivityNode\(EveningActivityMovement\)**](#) - Method in class

movement.[EveningActivityControlSystem](#)

Register a evening activity node with the system

[**addHeading\(String\)**](#) - Method in class gui.[EventLogControlPanel](#)

Adds a new heading in the control panel.

[**addInterface\(NetworkInterface\)**](#) - Method in class interfaces.[ConnectivityGrid](#)

Adds a network interface to the overlay grid

[**addInterface\(NetworkInterface\)**](#) - Method in class interfaces.[ConnectivityGrid.GridCell](#)

Adds an interface to this cell

[**addInterface\(NetworkInterface\)**](#) - Method in class interfaces.[ConnectivityOptimizer](#)

Adds a network interface to the optimizer (unless it is already present)

[**addInterfaces\(Collection<NetworkInterface>\)**](#) - Method in class interfaces.[ConnectivityGrid](#)

Adds interfaces to overlay grid

[**addInterfaces\(Collection<NetworkInterface>\)**](#) - Method in class interfaces.[ConnectivityOptimizer](#)

Adds a collection of network interfaces to the optimizer (except of those already added)

[**addMessageListener\(MessageListener\)**](#) - Method in class core.[SimScenario](#)

Adds a new message listener for all nodes

[**addMessageTransfer\(DTNHost, DTNHost\)**](#) - Method in class gui.playfield.[PlayField](#)

Adds graphics for message transfer

[**addMoreInfo\(RoutingInfo\)**](#) - Method in class routing.[RoutingInfo](#)

Adds child info object for this routing info.

[**addMovementListener\(MovementListener\)**](#) - Method in class core.[SimScenario](#)

Adds a new movement listener for all nodes

[**addNeighbor\(MapNode\)**](#) - Method in class movement.map.[MapNode](#)

Adds the node as this node's neighbour (unless the node is null)

[**addNode\(EveningActivityMovement\)**](#) - Method in class movement.[EveningTrip](#)

Add an evening activity node to the group

[**addNodeOnPath\(DTNHost\)**](#) - Method in class core.[Message](#)

Adds a new node on the list of nodes this message has passed

[**addPath\(Path\)**](#) - Method in class gui.playfield.[PlayField](#)

Adds a path to the overlay graphics

[**addPaths\(File, int\)**](#) - Method in class input.[WKTMapReader](#)

Adds paths to the map and adds given type to all nodes' type.

[**addPaths\(Reader, int\)**](#) - Method in class input.[WKTMapReader](#)

Add paths to current path set.

[**addProperty\(String, Object\)**](#) - Method in class core.[Message](#)

Adds a generic property for this message.

[**addProperty\(String, Object\)**](#) - Method in class core.[ModuleCommunicationBus](#)

Adds a new property for this node.

[**addReport\(Report\)**](#) - Method in class ui.[DTNSimUI](#)

Adds a new report for simulator

[**addSettings\(String\)**](#) - Static method in class core.[Settings](#)

Reads another settings file and adds the key-value pairs to the current settings overriding any values that already existed with the same keys.

[**addToMessages\(Message, boolean\)**](#) - Method in class routing.[MessageRouter](#)

Adds a message to the message buffer and informs message listeners about new message (if requested).

[**addToSendingConnections\(Connection\)**](#) - Method in class routing.[ActiveRouter](#)

Adds a connections to sending connections which are monitored in the update.

[**addType\(int\)**](#) - Method in class movement.map.[MapNode](#)

Adds a type indicator to this node

[**addUpdate\(double\)**](#) - Method in class input.[ScheduledUpdatesQueue](#)

Add a new update request for the given time

[**addUpdateListener\(UpdateListener\)**](#) - Method in class core.[SimScenario](#)

Adds a new update listener for the world

[**addWarmupID\(String\)**](#) - Method in class report.[Report](#)

Adds a new ID to the warm up ID set

[**addWaypoint\(Coord\)**](#) - Method in class movement.[Path](#)

Adds a new waypoint to the end of the path.

[**addWaypoint\(Coord, double\)**](#) - Method in class movement.[Path](#)

Adds a new waypoint with a speed towards that waypoint

[**AdjacencyGraphvizReport**](#) - Class in [report](#)

Generates Graphviz compatible graph from connections.

[**AdjacencyGraphvizReport\(\)**](#) - Constructor for class report.[AdjacencyGraphvizReport](#)

Constructor.

[**advance\(double\)**](#) - Method in class core.[SimClock](#)

Advances the time by n seconds

[**ALL_MESSAGES_ID**](#) - Static variable in class input.[StandardEventsReader](#)

Message identifier to use to refer to all messages ("*")

[**allMembersPresent\(\)**](#) - Method in class movement.[EveningTrip](#)

Checks if all members of the group have found their way to the meeting point

[**ALPHA_S**](#) - Static variable in class routing.[MaxPropRouter](#)

The alpha parameter string

[**APP_ID**](#) - Static variable in class applications.[PingApplication](#)

Application ID

[**APPCOUNT_S**](#) - Static variable in class core.[SimScenario](#)

setting name for the number of applications

[**appID**](#) - Variable in class core.[Application](#)

[**Application**](#) - Class in [core](#)

Base class for applications.

[**Application\(\)**](#) - Constructor for class core.[Application](#)

[**Application\(Application\)**](#) - Constructor for class core.[Application](#)

Copy constructor.

[**ApplicationListener**](#) - Interface in [core](#)

Interface for classes that want to be informed about messages between hosts.

[**applications**](#) - package applications

[**APPTYPE_NS**](#) - Static variable in class core.[SimScenario](#)

namespace for application type settings ("Application")

[**APPTYPE_S**](#) - Static variable in class core.[SimScenario](#)

application type -setting id ("type")

[**assertValidRange\(int\[\], String\)**](#) - Method in class core.[Settings](#)

Checks that the given integer array contains a valid range.

B

[**B_SIZE_S**](#) - Static variable in class routing.[MessageRouter](#)

Message buffer size -setting id ("bufferSize").

[**backAllowed**](#) - Variable in class movement.[MapBasedMovement](#)

May a node choose to move back the same way it came at a crossing

[**BATCH_MODE_FLAG**](#) - Static variable in class core.[DTNSim](#)

If this option ("-b") is given to program, batch mode and Text UI are used

[**BETA_S**](#) - Static variable in class routing.[ProphetRouter](#)

Transitivity scaling constant (beta) -setting id ("beta").

[**BETA_S**](#) - Static variable in class routing.[ProphetRouterWithEstimation](#)

Transitivity scaling constant (beta) -setting id ("beta").

[**BINARY_EXT**](#) - Static variable in class input.[BinaryEventsReader](#)

Extension of binary external events file

[**BINARY_MODE**](#) - Static variable in class routing.[SprayAndWaitRouter](#)

identifier for the binary-mode setting ("binaryMode")

[**BinaryEventsReader**](#) - Class in [input](#)

Reads External Events from a binary file.

[**BinaryEventsReader\(File\)**](#) - Constructor for class input.[BinaryEventsReader](#)

Constructor.

[**BUS CONTROL SYSTEM NR**](#) - Static variable in class movement.[BusControlSystem](#)

[**BusControlSystem**](#) - Class in [movement](#)

This class controls busses and passengers that can use the bus.

[**busHasStopped\(int, Coord, Path\)**](#) - Method in class movement.[BusControlSystem](#)

Called by busses belonging to this system every time the bus has stopped.

[**BusMovement**](#) - Class in [movement](#)

This class controls the movement of busses.

[**BusMovement\(Settings\)**](#) - Constructor for class movement.[BusMovement](#)

Creates a new instance of BusMovement

[**BusMovement\(BusMovement\)**](#) - Constructor for class movement.[BusMovement](#)

Create a new instance from a prototype

[**BusTravellerMovement**](#) - Class in [movement](#)

This class controls the movement of bus travellers.

[**BusTravellerMovement\(Settings\)**](#) - Constructor for class movement.[BusTravellerMovement](#)

Creates a BusTravellerModel

[**BusTravellerMovement\(BusTravellerMovement\)**](#) - Constructor for class movement.[BusTravellerMovement](#)

Creates a BusTravellerModel from a prototype

[**BYTES_TRANSFERRED_AVG_SAMPLES**](#) - Static variable in class routing.[MaxPropRouter](#)

Over how many samples the "average number of bytes transferred per transfer opportunity" is taken

[**BYTES_TRANSFERRED_AVG_SAMPLES**](#) - Static variable in class routing.[MaxPropRouterWithEstimation](#)

Over how many samples the "average number of bytes transferred per transfer opportunity" is taken

[**bytesTransferred**](#) - Variable in class core.[Connection](#)

how many bytes this connection has transferred

C

[**calcThreshold\(\)**](#) - Method in class routing.[MaxPropRouter](#)

Calculates and returns the current threshold value for the buffer's split based on the average number of bytes transferred per transfer opportunity and the hop counts of the messages in the buffer.

[**calcThreshold\(\)**](#) - Method in class routing.[MaxPropRouterWithEstimation](#)

Calculates and returns the current threshold value for the buffer's split based on the average number of bytes transferred per transfer opportunity and the hop counts of the messages in the buffer.

[**cancelSim\(\)**](#) - Method in class core.[World](#)

Asynchronously cancels the currently running simulation

[**canStartTransfer\(\)**](#) - Method in class routing.[ActiveRouter](#)

Makes rudimentary checks (that we have at least one message and one connection) about can this router start transfer.

[**CarMovement**](#) - Class in [movement](#)

The CarMovement class representing the car movement submodel

[**CarMovement\(Settings\)**](#) - Constructor for class movement.[CarMovement](#)

Car movement constructor

[**CarMovement\(CarMovement\)**](#) - Constructor for class movement.[CarMovement](#)

Construct a new CarMovement instance from a prototype

[**CBRConnection**](#) - Class in [core](#)

A constant bit-rate connection between two DTN nodes.

[**CBRConnection\(DTNHost, NetworkInterface, DTNHost, NetworkInterface, int\)**](#) - Constructor for class core.[CBRConnection](#)

Creates a new connection between nodes and sets the connection state to "up".

[**CELL_SIZE_MULT_S**](#) - Static variable in class core.[World](#)

Cell based optimization cell size multiplier -setting id ("cellSizeMult").

[**centerViewAt\(Coord\)**](#) - Method in class gui.[DTNSimGUI](#)

Sets certain location to be in the center of the playfield view

[**changedConnection\(Connection\)**](#) - Method in class routing.[ActiveRouter](#)

Called when a connection's state changes.

[**changedConnection\(Connection\)**](#) - Method in class routing.[EpidemicOracleRouter](#)[**changedConnection\(Connection\)**](#) - Method in class routing.[MaxPropRouter](#)[**changedConnection\(Connection\)**](#) - Method in class routing.[MaxPropRouterWithEstimation](#)[**changedConnection\(Connection\)**](#) - Method in class routing.[MessageRouter](#)

Informs the router about change in connections state.

[**changedConnection\(Connection\)**](#) - Method in class routing.[PassiveRouter](#)[**changedConnection\(Connection\)**](#) - Method in class routing.[ProphetRouter](#)[**changedConnection\(Connection\)**](#) - Method in class routing.[ProphetRouterWithEstimation](#)[**changeZoom\(int\)**](#) - Method in class gui.[GUIControls](#)

Changes the zoom level

[**checkReceiving\(Message\)**](#) - Method in class routing.[ActiveRouter](#)

Checks if router "wants" to start receiving message (i.e.

[**checkReceiving\(Message\)**](#) - Method in class routing.[EnergyAwareRouter](#)[**checkReceiving\(Message\)**](#) - Method in class routing.[EpidemicOracleRouter](#)[**checkReceiving\(Message\)**](#) - Method in class routing.[FirstContactRouter](#)[**CIRCULAR**](#) - Static variable in class movement.map.[MapRoute](#)

Type of the route ID: circular (1).

[**CLASS_PACKAGE**](#) - Static variable in class input.[EventQueueHandler](#)

name of the package where event generator classes are looked from

[**CLASS_SETTING**](#) - Static variable in class input.[EventQueueHandler](#)

name of the events class (for class based events) -setting id ("class")

[**clearMsgOnFly\(\)**](#) - Method in class core.[Connection](#)

Clears the message that is currently being transferred.

[**clearOverlays\(\)**](#) - Method in class gui.playfield.[PlayField](#)

Removes all overlay graphics stored to be drawn

[**clone\(\)**](#) - Method in class core.[Coord](#)

Returns a clone of this coordinate

[**close\(\)**](#) - Method in class input.[BinaryEventsReader](#)[**close\(\)**](#) - Method in interface input.[ExternalEventsReader](#)

Closes the input file streams of the reader.

[**close\(\)**](#) - Method in class input.[StandardEventsReader](#)[**closeSim\(\)**](#) - Method in class gui.[DTNSimGUI](#)

Closes the program if simulation is done or cancels it.

[**CLUSTER_CENTER**](#) - Static variable in class movement.[ClusterMovement](#)

Center point of the cluster

[**CLUSTER RANGE**](#) - Static variable in class movement.[ClusterMovement](#)

Range of the cluster

[**ClusterMovement**](#) - Class in [movement](#)

[**ClusterMovement\(Settings\)**](#) - Constructor for class movement.[ClusterMovement](#)

[**comBus**](#) - Variable in class movement.[MovementModel](#)

[**COMMENT PREFIX**](#) - Static variable in class input.[ExternalMovementReader](#)

[**compareByQueueMode\(Message, Message\)**](#) - Method in class routing.[MessageRouter](#)

Gives the order of the two given messages as defined by the current queue mode

[**compareTo\(Coord\)**](#) - Method in class core.[Coord](#)

Compares this coordinate to other coordinate.

[**compareTo\(DTNHost\)**](#) - Method in class core.[DTNHost](#)

Compares two DTNHosts by their addresses.

[**compareTo\(Message\)**](#) - Method in class core.[Message](#)

Compares two messages by their ID (alphabetically).

[**compareTo\(ExternalEvent\)**](#) - Method in class input.[ExternalEvent](#)

Compares two external events by their time.

[**compareTo\(MapNode\)**](#) - Method in class movement.map.[MapNode](#)

Compares two map nodes by their coordinates

[**connect\(DTNHost\)**](#) - Method in class core.[DTNHost](#)

for tests only --- do not use!!!

[**connect\(NetworkInterface\)**](#) - Method in class core.[NetworkInterface](#)

Connects the interface to another interface.

[**connect\(Connection, NetworkInterface\)**](#) - Method in class core.[NetworkInterface](#)

Connects this host to another host.

[**connect\(NetworkInterface\)**](#) - Method in class interfaces.[InterferenceLimitedInterface](#)

Tries to connect this host to another host.

[**connect\(NetworkInterface\)**](#) - Method in class interfaces.[SimpleBroadcastInterface](#)

Tries to connect this host to another host.

[**connected\(\)**](#) - Method in class input.[DTN2Events.ParserHandler](#)

[**Connection**](#) - Class in [core](#)

A connection between two DTN nodes.

[**Connection\(DTNHost, NetworkInterface, DTNHost, NetworkInterface\)**](#) - Constructor for class core.[Connection](#)

Creates a new connection between nodes and sets the connection state to "up".

[**CONNECTION**](#) - Static variable in class input.[StandardEventsReader](#)

Identifier of connection event ("CONN")

[**CONNECTION DOWN**](#) - Static variable in class input.[StandardEventsReader](#)

Value identifier of connection down event ("down")

[**CONNECTION UP**](#) - Static variable in class input.[StandardEventsReader](#)

Value identifier of connection up event ("up")

[**connectionDown\(Connection\)**](#) - Method in class core.[DTNHost](#)

[**connectionEnd\(\)**](#) - Method in class report.[ContactTimesReport.ConnectionInfo](#)

Should be called when the connection ended to record the time.

[**ConnectionEvent**](#) - Class in [input](#)

A connection up/down event.

[**ConnectionEvent\(int, int, String, boolean, double\)**](#) - Constructor for class input.[ConnectionEvent](#)

Creates a new connection event

[**ConnectionListener**](#) - Interface in [core](#)

Interface for classes that want to be informed about connections between hosts.

[**connections**](#) - Variable in class core.[NetworkInterface](#)

[**connections**](#) - Variable in class report.[ContactTimesReport](#)

[**connectionUp\(Connection\)**](#) - Method in class core.[DTNHost](#)

Informs the router of this host about state change in a connection object.

[**ConnectivityDtnsim2Report**](#) - Class in [report](#)

Link connectivity report generator for DTNSim2 input.

[**ConnectivityDtnsim2Report\(\)**](#) - Constructor for class report.[ConnectivityDtnsim2Report](#)

Constructor.

[**ConnectivityGrid**](#) - Class in [interfaces](#)

Overlay grid of the world where each interface is put on a cell depending of its location.

[**ConnectivityGrid.GridCell**](#) - Class in [interfaces](#)

A single cell in the cell grid.

[**ConnectivityGridFactory\(int, double\)**](#) - Static method in class interfaces.[ConnectivityGrid](#)

Returns a connectivity grid object based on a hash value

[**ConnectivityONEReport**](#) - Class in [report](#)

Link connectivity report generator for ONE StandardEventsReader input.

[**ConnectivityONEReport\(\)**](#) - Constructor for class report.[ConnectivityONEReport](#)

Constructor.

[**ConnectivityOptimizer**](#) - Class in [interfaces](#)

A superclass for schemes for optimizing the location of possible contacts with network interfaces of a specific range

[**ConnectivityOptimizer\(\)**](#) - Constructor for class interfaces.[ConnectivityOptimizer](#)

[**ContactsDuringAnICTReport**](#) - Class in [report](#)

The number of contacts during an inter-contact time metric is similar to the inter-contact times metric, except that instead of measuring the time until a node meets again, we count the number of other nodes both of the nodes meet separately.

[**ContactsDuringAnICTReport\(\)**](#) - Constructor for class report.[ContactsDuringAnICTReport](#)

[**ContactsPerHourReport**](#) - Class in [report](#)

This report counts the number of contacts each hour

[**ContactsPerHourReport\(\)**](#) - Constructor for class report.[ContactsPerHourReport](#)

[**ContactTimesReport**](#) - Class in [report](#)

Reports the node contact time (i.e., how long they were in the range of each other) distribution.

[**ContactTimesReport\(\)**](#) - Constructor for class report.[ContactTimesReport](#)

Constructor.

[**ContactTimesReport.ConnectionInfo**](#) - Class in [report](#)

Objects of this class store time information about contacts.

[**ContactTimesReport.ConnectionInfo\(DTNHost, DTNHost\)**](#) - Constructor for class

report.[ContactTimesReport.ConnectionInfo](#)

[**contains\(String\)**](#) - Method in class core.[Settings](#)

Returns true if a setting with defined name (in the current namespace or secondary namespace if such is set) exists and has some value (not just white space)

[**Coord**](#) - Class in [core](#)

Class to hold 2D coordinates and perform simple arithmetics and transformations

[**Coord\(double, double\)**](#) - Constructor for class core.[Coord](#)

Constructor.

[**COORD_FORMAT**](#) - Static variable in class report.[MovementNs2Report](#)

formatting string for coordinate values ("%.⁵f")

[**copyFrom\(Message\)**](#) - Method in class core.[Message](#)

Deep copies message data from other message.

[**core**](#) - package core

Contains core classes and interfaces of the simulator.

[**CREATE**](#) - Static variable in class input.[StandardEventsReader](#)

Identifier of message creation event ("C")

[**createConnection\(NetworkInterface\)**](#) - Method in class core.[NetworkInterface](#)

Creates a connection to another host.

[**createConnection\(NetworkInterface\)**](#) - Method in class interfaces.[InterferenceLimitedInterface](#)

Creates a connection to another host.

[**createConnection\(NetworkInterface\)**](#) - Method in class interfaces.[SimpleBroadcastInterface](#)

Creates a connection to another host.

[**CreatedMessagesReport**](#) - Class in [report](#)

Reports information about all created messages.

[**CreatedMessagesReport\(\)**](#) - Constructor for class report.[CreatedMessagesReport](#)

Constructor.

[**createHosts\(\)**](#) - Method in class core.[SimScenario](#)

Creates hosts for the scenario

[**createInitializedObject\(String\)**](#) - Method in class core.[Settings](#)

Creates (and dynamically loads the class of) an object that initializes itself using the settings in this Settings object (given as the only parameter to the constructor).

[**createNewMessage\(Message\)**](#) - Method in class core.[DTNHost](#)

Creates a new message to this host's router

[**createNewMessage\(Message\)**](#) - Method in class routing.[ActiveRouter](#)

[**createNewMessage\(Message\)**](#) - Method in class routing.[EpidemicOracleRouter](#)

[**createNewMessage\(Message\)**](#) - Method in class routing.[MessageRouter](#)

Creates a new message to the router.

[**createNewMessage\(Message\)**](#) - Method in class routing.[SprayAndWaitRouter](#)

[**createObject\(String\)**](#) - Method in class core.[Settings](#)

Creates (and dynamically loads the class of) an object using the constructor without any parameters.

[**currentTransmitSpeed**](#) - Variable in class interfaces.[InterferenceLimitedInterface](#)

D

[**Debug**](#) - Class in [core](#)

Debugging info printer with time stamping.

[**Debug\(\)**](#) - Constructor for class core.[Debug](#)

[**DEF_CON_CELL_SIZE_MULT**](#) - Static variable in class core.[World](#)

default value for cell size multiplier (5)

[**DEF_NODE_ARRAY**](#) - Static variable in class report.[MovementNs2Report](#)

default value for the array name ("\$node_")

[**DEF_NS_CMD**](#) - Static variable in class report.[MovementNs2Report](#)

default value for the ns command ("\$ns_")

[**DEF_PRECISION**](#) - Static variable in class report.[Report](#)

Default precision of formatted double values

[**DEF_RANDOMIZE_UPDATES**](#) - Static variable in class core.[World](#)

should the update order of nodes be randomized -setting's default value (true)

[**DEF_SETTINGS_FILE**](#) - Static variable in class core.[Settings](#)

file name of the default settings file ("default_settings.txt")

[**DEF_SPEEDS**](#) - Static variable in class movement.[MovementModel](#)

default setting for speed distribution

[**DEF_WAIT_TIMES**](#) - Static variable in class movement.[MovementModel](#)

default setting for wait time distribution

[**DEFAULT_ALPHA**](#) - Static variable in class routing.[MaxPropRouter](#)

The default value for alpha

[**DEFAULT_ALPHA**](#) - Static variable in class routing.[MaxPropRouterWithEstimation](#)

The default value for alpha

[**DEFAULT_BETA**](#) - Static variable in class routing.[ProphetRouter](#)

delivery predictability transitivity scaling constant default value

DEFAULT_BETA - Static variable in class routing.[ProphetRouterWithEstimation](#)

delivery predictability transitivity scaling constant default value

DEFAULT_NROF_PRELOAD - Static variable in class input.[ExternalEventsQueue](#)

default number of preloaded events

DEFAULT_PROB_SET_MAX_SIZE - Static variable in class routing.[MaxPropRouter](#)

Default value for the meeting probability set maximum size (50).

DEFAULT_PTARGET - Static variable in class routing.[ProphetRouterWithEstimation](#)

default P target

DELETE_DELIVERED_S - Static variable in class routing.[ActiveRouter](#)

Delete delivered messages -setting id ("deleteDelivered").

deleteDelivered - Variable in class routing.[ActiveRouter](#)

should messages that final recipient marks as delivered be deleted from message buffer

deleteMessage(String, boolean) - Method in class core.[DTNHost](#)

Deletes a message from this host

deleteMessage(String, boolean) - Method in class routing.[MessageRouter](#)

Deletes a message from the buffer and informs message listeners about the event

DELIVERED - Static variable in class input.[StandardEventsReader](#)

Identifier of message delivered event ("DE")

DeliveredMessagesReport - Class in [report](#)

Report information about all delivered messages.

DeliveredMessagesReport() - Constructor for class report.[DeliveredMessagesReport](#)

Constructor.

DENIED_NO_SPACE - Static variable in class routing.[MessageRouter](#)

Receive return value for not enough space in the buffer for the msg

DENIED_OLD - Static variable in class routing.[MessageRouter](#)

Receive return value for an old (already received) message

DENIED_TTL - Static variable in class routing.[MessageRouter](#)

Receive return value for messages whose TTL has expired

DENIED_UNSPECIFIED - Static variable in class routing.[MessageRouter](#)

Receive return value for unspecified reason

destroyConnection(NetworkInterface) - Method in class core.[NetworkInterface](#)

Disconnect a connection between this and another host.

DijkstraPathFinder - Class in [movement.map](#)

Implementation of the Dijkstra's shortest path algorithm.

DijkstraPathFinder(int[]) - Constructor for class movement.map.[DijkstraPathFinder](#)

Constructor.

DirectDeliveryRouter - Class in [routing](#)

Router that will deliver messages only to the final recipient.

DirectDeliveryRouter(Settings) - Constructor for class routing.[DirectDeliveryRouter](#)

DirectDeliveryRouter(DirectDeliveryRouter) - Constructor for class routing.[DirectDeliveryRouter](#)

disconnect(Connection, NetworkInterface) - Method in class core.[NetworkInterface](#)

Disconnects this host from another host.

distance(Coord) - Method in class core.[Coord](#)

Returns the distance to another coordinate

DistanceDelayReport - Class in [report](#)

Report for how far apart the nodes were when the message was sent and how long time & how many hops it took to deliver it.

DistanceDelayReport() - Constructor for class report.[DistanceDelayReport](#)

Constructor.

done() - Method in class report.[AdjacencyGraphvizReport](#)

done() - Method in class report.[ContactsDuringAnICTReport](#)

done() - Method in class report.[ContactsPerHourReport](#)

[**done\(\)**](#) - Method in class report.[ContactTimesReport](#)

[**done\(\)**](#) - Method in class report.[CreatedMessagesReport](#)

[**done\(\)**](#) - Method in class report.[DeliveredMessagesReport](#)

[**done\(\)**](#) - Method in class report.[DistanceDelayReport](#)

[**done\(\)**](#) - Method in class report.[EncountersVSUniqueEncountersReport](#)

[**done\(\)**](#) - Method in class report.[MessageDelayReport](#)

[**done\(\)**](#) - Method in class report.[MessageDeliveryReport](#)

[**done\(\)**](#) - Method in class report.[MessageGraphvizReport](#)

[**done\(\)**](#) - Method in class report.[MessageReport](#)

[**done\(\)**](#) - Method in class report.[MessageStatsReport](#)

[**done\(\)**](#) - Method in class report.[PingAppReporter](#)

[**done\(\)**](#) - Method in class report.[Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

[**done\(\)**](#) - Method in class report.[TotalEncountersReport](#)

[**done\(\)**](#) - Method in class report.[UniqueEncountersReport](#)

[**done\(\)**](#) - Method in class ui.[DTNSimUI](#)

Runs maintenance jobs that are needed before exiting.

[**doneTiming\(\)**](#) - Static method in class core.[Debug](#)

End timing an action.

[**draw\(Graphics2D\)**](#) - Method in class gui.playfield.[MapGraphic](#)

[**draw\(Graphics2D\)**](#) - Method in class gui.playfield.[MessageGraphic](#)

[**draw\(Graphics2D\)**](#) - Method in class gui.playfield.[NodeGraphic](#)

[**draw\(Graphics2D\)**](#) - Method in class gui.playfield.[PathGraphic](#)

Draws a line trough all path's coordinates.

[**draw\(Graphics2D\)**](#) - Method in class gui.playfield.[PlayFieldGraphic](#)

Draws the graphic component to the graphics context g2

[**draw\(Graphics2D\)**](#) - Method in class gui.playfield.[ScaleReferenceGraphic](#)

[**drawHostAddress\(int\[\]\)**](#) - Method in class input.[MessageEventGenerator](#)

Draws a random host address from the configured address range

[**drawMessageSize\(\)**](#) - Method in class input.[MessageEventGenerator](#)

Generates a (random) message size

[**drawNextEventTimeDiff\(\)**](#) - Method in class input.[MessageEventGenerator](#)

Generates a (random) time difference between two events

[**drawToAddress\(int\[\], int\)**](#) - Method in class input.[MessageEventGenerator](#)

Draws a destination host address that is different from the "from" address

[**DROP**](#) - Static variable in class input.[StandardEventsReader](#)

Identifier of message dropped event ("DR")

[**dropExpiredMessages\(\)**](#) - Method in class routing.[ActiveRouter](#)

Drops messages whose TTL is less than zero.

[DTN2Events](#) - Class in [input](#)

Delivers bundles from dtnd to ONE.

[DTN2Events\(Settings\)](#) - Constructor for class [input.DTN2Events](#)

Creates a new events object.

[DTN2Events.ParserHandler](#) - Class in [input](#)

Inner class that implements the CLA interface for receiving bundles from dtnd.

[DTN2Events.ParserHandler\(int, DTN2Events, String, int\)](#) - Constructor for class [input.DTN2Events.ParserHandler](#)

Creates a new parser handler.

[DTN2Manager](#) - Class in [core](#)

Manages the external convergence layer connections to dtnd.

[DTN2Manager\(\)](#) - Constructor for class [core.DTN2Manager](#)

[DTN2Manager.EIDHost](#) - Class in [core](#)

EID to DTNHost mapping elements.

[DTN2Manager.EIDHost\(String, int, DTNHost\)](#) - Constructor for class [core.DTN2Manager.EIDHost](#)

[DTN2Reporter](#) - Class in [report](#)

The DTN2Reporter class is responsible for delivering bundles from The ONE to dtnd.

[DTN2Reporter\(\)](#) - Constructor for class [report.DTN2Reporter](#)

Creates a new reporter object.

[DTNHost](#) - Class in [core](#)

A DTN capable host.

[DTNHost\(List<MessageListener>, List<MovementListener>, String, List<NetworkInterface>, ModuleCommunicationBus, MovementModel, MessageRouter\)](#) - Constructor for class [core.DTNHost](#)

Creates a new DTNHost.

[DTNSim](#) - Class in [core](#)

Simulator's main class

[DTNSim\(\)](#) - Constructor for class [core.DTNSim](#)

[DTNSimGUI](#) - Class in [gui](#)

Graphical User Interface for simulator

[DTNSimGUI\(\)](#) - Constructor for class [gui.DTNSimGUI](#)

[DTNSimTextUI](#) - Class in [ui](#)

Simple text-based user interface.

[DTNSimTextUI\(\)](#) - Constructor for class [ui.DTNSimTextUI](#)

[DTNSimUI](#) - Class in [ui](#)

Abstract superclass for user interfaces; contains also some simulation settings.

[DTNSimUI\(\)](#) - Constructor for class [ui.DTNSimUI](#)

Constructor.

E

[EID](#) - Variable in class [core.DTN2Manager.EIDHost](#)

[EncountersVSUniqueEncountersReport](#) - Class in [report](#)

The total- vs.

[EncountersVSUniqueEncountersReport\(\)](#) - Constructor for class [report.EncountersVSUniqueEncountersReport](#)

[END_LOCATION_S](#) - Static variable in class [movement.LinearFormation](#)

Per node group setting for defining the end coordinates of the line ("endLocation")

[END_TIME_S](#) - Static variable in class [core.SimScenario](#)

end time -setting id ("endTime")

[**ENERGY_VALUE_ID**](#) - Static variable in class routing.[EnergyAwareRouter](#)

[**ModuleCommunicationBus**](#) identifier for the "current amount of energy left" variable.

[**EnergyAwareRouter**](#) - Class in [routing](#)

Energy level-aware variant of Epidemic router.

[**EnergyAwareRouter\(Settings\)**](#) - Constructor for class routing.[EnergyAwareRouter](#)

Constructor.

[**EnergyAwareRouter\(EnergyAwareRouter\)**](#) - Constructor for class routing.[EnergyAwareRouter](#)

Copy constructor.

[**EnergyLevelReport**](#) - Class in [report](#)

Node energy level report.

[**EnergyLevelReport\(\)**](#) - Constructor for class report.[EnergyLevelReport](#)

Constructor.

[**ensurePositiveValue\(double, String\)**](#) - Method in class core.[NetworkInterface](#)

Makes sure that a value is positive

[**enterBus\(Path\)**](#) - Method in class movement.[BusTravellerMovement](#)

Notifies the node at the bus stop that a bus is there.

[**EpidemicOracleRouter**](#) - Class in [routing](#)

Epidemic message router with an oracle that tells when a message is delivered and that message is then removed from all nodes that use this routing module.

[**EpidemicOracleRouter\(Settings\)**](#) - Constructor for class routing.[EpidemicOracleRouter](#)

Constructor.

[**EpidemicOracleRouter\(EpidemicOracleRouter\)**](#) - Constructor for class routing.[EpidemicOracleRouter](#)

Copy constructor.

[**EpidemicRouter**](#) - Class in [routing](#)

Epidemic message router with drop-oldest buffer and only single transferring connections at a time.

[**EpidemicRouter\(Settings\)**](#) - Constructor for class routing.[EpidemicRouter](#)

Constructor.

[**EpidemicRouter\(EpidemicRouter\)**](#) - Constructor for class routing.[EpidemicRouter](#)

Copy constructor.

[**EPSILON**](#) - Static variable in class report.[MovementNs2Report](#)

a value "close enough" to zero (1.0E-5).

[**equals\(Coord\)**](#) - Method in class core.[Coord](#)

Checks if this coordinate's location is equal to other coordinate's

[**equals\(Object\)**](#) - Method in class core.[Coord](#)

[**equals\(DTNHost\)**](#) - Method in class core.[DTNHost](#)

Checks if a host is the same as this host by comparing the object reference

[**equals\(Object\)**](#) - Method in class report.[ContactTimesReport.ConnectionInfo](#)

Returns true if the other connection info contains the same hosts.

[**error\(String, Exception, boolean\)**](#) - Method in class input.[DTN2Events.ParserHandler](#)

[**EVENING_ACTIVITY_CONTROL_SYSTEM_NR_SETTING**](#) - Static variable in class

movement.[EveningActivityMovement](#)

[**EveningActivityControlSystem**](#) - Class in [movement](#)

This class controls the group mobility of the people meeting their friends in the evening

[**EveningActivityMovement**](#) - Class in [movement](#)

A Class to model movement when people are out shopping or doing other activities with friends.

[**EveningActivityMovement\(Settings\)**](#) - Constructor for class movement.[EveningActivityMovement](#)

Creates a new instance of EveningActivityMovement

[**EveningActivityMovement\(EveningActivityMovement\)**](#) - Constructor for class

movement.[EveningActivityMovement](#)

Creates a new instance of EveningActivityMovement from a prototype

[**EveningTrip**](#) - Class in [movement](#)

A class to encapsulate information about a shopping trip 1.

[**EveningTrip\(int, Coord\)**](#) - Constructor for class movement.[EveningTrip](#)

Create a new instance of a EveningTrip

[EventLogControl](#) - Class in [gui](#)

Class encapsulates the references to the controls one can add to the EventLogControlPanel

[EventLogControl\(JCheckBox, JCheckBox\)](#) - Constructor for class [gui.EventLogControl](#)
Constructor.[EventLogControlPanel](#) - Class in [gui](#)

Control panel for event log

[EventLogControlPanel\(\)](#) - Constructor for class [gui.EventLogControlPanel](#)
Constructor.[EventLogPanel](#) - Class in [gui](#)

Event log panel where log entries are displayed.

[EventLogPanel\(DTNSimGUI\)](#) - Constructor for class [gui.EventLogPanel](#)
Creates a new log panel[EventLogReport](#) - Class in [report](#)

Report that creates same output as the GUI's event log panel but formatted like [StandardEventsReader](#) input.

[EventLogReport\(\)](#) - Constructor for class [report.EventLogReport](#)[EventQueue](#) - Interface in [input](#)

Interface for event queues.

[EventQueueHandler](#) - Class in [input](#)

Handler for managing event queues.

[EventQueueHandler\(\)](#) - Constructor for class [input.EventQueueHandler](#)

Creates a new EventQueueHandler which can be queried for event queues.

[eventsLeftInBuffer\(\)](#) - Method in class [input.ExternalEventsQueue](#)

Returns the amount of events left in the buffer at the moment (the amount can increase later if more events are read).

[exchangeDeliverableMessages\(\)](#) - Method in class [routing.ActiveRouter](#)

Exchanges deliverable (to final recipient) messages between this host and all hosts this host is currently connected to.

[ExtendedMovementModel](#) - Class in [movement](#)

Classes derived from this can make use of other movement models that implement the SwitchableMovement interface.

[ExtendedMovementModel\(\)](#) - Constructor for class [movement.ExtendedMovementModel](#)

Creates a new ExtendedMovementModel

[ExtendedMovementModel\(Settings\)](#) - Constructor for class [movement.ExtendedMovementModel](#)

Creates a new ExtendedMovementModel

[ExtendedMovementModel\(ExtendedMovementModel\)](#) - Constructor for class [movement.ExtendedMovementModel](#)

Creates a new ExtendedMovementModel from a prototype

[EXTERNAL_MOVEMENT_NS](#) - Static variable in class [movement.ExternalMovement](#)

Namespace for settings

[ExternalEvent](#) - Class in [input](#)

Super class for all external events.

[ExternalEvent\(double\)](#) - Constructor for class [input.ExternalEvent](#)[ExternalEventsQueue](#) - Class in [input](#)

Queue of external events.

[ExternalEventsQueue\(String, int\)](#) - Constructor for class [input.ExternalEventsQueue](#)

Creates a new Queue from a file

[ExternalEventsQueue\(Settings\)](#) - Constructor for class [input.ExternalEventsQueue](#)

Create a new Queue based on the given settings: [ExternalEventsQueue.PRELOAD_SETTING](#) and [ExternalEventsQueue.PATH_SETTING](#).

[ExternalEventsReader](#) - Interface in [input](#)

Interface for external event readers.

[ExternalMovement](#) - Class in [movement](#)

Movement model that uses external data of node locations.

[ExternalMovement\(Settings\)](#) - Constructor for class [movement.ExternalMovement](#)

Constructor for the prototype.

[ExternalMovementReader](#) - Class in [input](#)

Reader for ExternalMovement movement model's time-location tuples.

[ExternalMovementReader\(String\)](#) - Constructor for class [input.ExternalMovementReader](#)
Constructor.

F

[FFW SPEED INDEX](#) - Static variable in class [gui.GUIControls](#)
index of FFW speed setting

[FILE_S](#) - Static variable in class [movement.MapBasedMovement](#)
map file -setting id ("mapFile")

[FILL_DELIMITER](#) - Static variable in class [core.Settings](#)
delimiter for requested values in strings ("% %")

[finalizeTransfer\(\)](#) - Method in class [core.Connection](#)
Finalizes the transfer of the currently transferred message.

[FirstContactRouter](#) - Class in [routing](#)
First contact router which uses only a single copy of the message (or fragments) and forwards it to the first available contact.

[FirstContactRouter\(Settings\)](#) - Constructor for class [routing](#).[FirstContactRouter](#)
Constructor.

[FirstContactRouter\(FirstContactRouter\)](#) - Constructor for class [routing](#).[FirstContactRouter](#)
Copy constructor.

[forceConnection\(DTNHost, String, boolean\)](#) - Method in class [core.DTNHost](#)
Force a connection event

[format\(double\)](#) - Method in class [report.Report](#)
Formats a double value according to current precision setting (see [Report.PRECISION_SETTING](#)) and returns it in a string.

[fromAddr](#) - Variable in class [input.ConnectionEvent](#)
address of the node the (dis)connection is from

[fromAddr](#) - Variable in class [input.MessageEvent](#)
address of the node the message is from

[fromInterface](#) - Variable in class [core.Connection](#)

[fromNode](#) - Variable in class [core.Connection](#)

G

[GAMMA](#) - Static variable in class [routing.ProphetRouter](#)
delivery predictability aging constant

[GAMMA](#) - Static variable in class [routing.ProphetRouterWithEstimation](#)
delivery predictability aging constant

[GAPPNAME_S](#) - Static variable in class [core.SimScenario](#)
application name in the group -setting id ("application")

[generateSpeed\(\)](#) - Method in class [movement.MovementModel](#)
Generates and returns a speed value between min and max of the [MovementModel.WAIT_TIME](#) setting.

[generateWaitTime\(\)](#) - Method in class [movement.BusTravellerMovement](#)
Switches state between getPath() calls

[generateWaitTime\(\)](#) - Method in class [movement.EveningActivityMovement](#)

[generateWaitTime\(\)](#) - Method in class [movement.ExtendedMovementModel](#)

[generateWaitTime\(\)](#) - Method in class [movement.HomeActivityMovement](#)

[**generateWaitTime\(\)**](#) - Method in class movement.[MovementModel](#)

Generates and returns a suitable waiting time at the end of a path.

[**generateWaitTime\(\)**](#) - Method in class movement.[OfficeActivityMovement](#)

[**getAddress\(\)**](#) - Method in class core.[DTNHost](#)

Returns the network-layer address of this host.

[**getAddress\(\)**](#) - Method in class core.[NetworkInterface](#)

Returns the network interface address.

[**getAllInterfaces\(\)**](#) - Method in class interfaces.[ConnectivityGrid](#)

Returns all interfaces that use the same technology and channel

[**getAllInterfaces\(\)**](#) - Method in class interfaces.[ConnectivityOptimizer](#)

Finds all other interfaces that are registered to the ConnectivityOptimizer

[**getAllProbs\(\)**](#) - Method in class routing.maxprop.[MeetingProbabilitySet](#)

Returns a reference to the probability map of this probability set

[**getAppID\(\)**](#) - Method in class core.[Application](#)

Returns an unique application ID.

[**getAppID\(\)**](#) - Method in class core.[Message](#)

[**getApplicationListeners\(\)**](#) - Method in class core.[SimScenario](#)

Returns the list of registered application event listeners

[**getApplications\(String\)**](#) - Method in class routing.[MessageRouter](#)

Returns all the applications that want to receive messages for the given application ID.

[**getAppListeners\(\)**](#) - Method in class core.[Application](#)

[**getAverage\(List<Double>\)**](#) - Method in class report.[Report](#)

Returns the average of double values stored in a List or "NaN" for empty lists.

[**getBoolean\(String\)**](#) - Method in class core.[Settings](#)

Returns a boolean-valued setting

[**getBufferOccupancy\(\)**](#) - Method in class core.[DTNHost](#)

Returns the buffer occupancy percentage.

[**getBufferSize\(\)**](#) - Method in class routing.[MessageRouter](#)

Returns the size of the message buffer.

[**getBundle\(String\)**](#) - Static method in class core.[DTN2Manager](#)

Returns the bundle associated with the given message id.

[**getBusControlSystem\(int\)**](#) - Static method in class movement.[BusControlSystem](#)

Returns a reference to a BusControlSystem with ID provided as parameter.

[**getBusStops\(\)**](#) - Method in class movement.[BusControlSystem](#)

[**getCenterViewCoord\(\)**](#) - Method in class gui.[DTNSimGUI](#)

Returns the world coordinates that are currently in the center of the viewport

[**getComBus\(\)**](#) - Method in class core.[DTNHost](#)

Returns this hosts's ModuleCommunicationBus

[**getComBus\(\)**](#) - Method in class movement.[MovementModel](#)

Returns the module communication bus of this movement model (if any)

[**getConnected\(int, double\)**](#) - Method in class routing.schedule.[ScheduleOracle](#)

Returns a list of schedule entries for nodes reachable after given time from the given node

[**getConnections\(\)**](#) - Method in class core.[DTNHost](#)

Returns a copy of the list of connections this host has with other hosts

[**getConnections\(\)**](#) - Method in class core.[NetworkInterface](#)

Returns a list of currently connected connections

[**getConnections\(\)**](#) - Method in class routing.[ActiveRouter](#)

Returns a list of connections this host currently has with other hosts.

[**getConnectionTime\(\)**](#) - Method in class report.[ContactTimesReport.ConnectionInfo](#)

Returns the time that passed between creation of this info and call to

[ContactTimesReport.ConnectionInfo.connectionEnd\(\)](#).

[**getControls\(\)**](#) - Method in class gui.[EventLogPanel](#)

Returns the control panel that this log uses

[getCoords\(\)](#) - Method in class movement.[Path](#)

Returns a reference to the coordinates of this path

[getCost\(DTNHost, DTNHost\)](#) - Method in class routing.[MaxPropRouter](#)

Returns the message delivery cost between two hosts from this host's point of view.

[getCost\(DTNHost, DTNHost\)](#) - Method in class routing.[MaxPropRouterWithEstimation](#)

Returns the message delivery cost between two hosts from this host's point of view.

[getCosts\(Integer, Set<Integer>\)](#) - Method in class routing.maxprop.[MaxPropDijkstra](#)

Calculates total costs to the given set of target nodes.

[getCreationTime\(\)](#) - Method in class core.[Message](#)

Returns the time when this message was created

[getCsvDoubles\(String, int\)](#) - Method in class core.[Settings](#)

Returns an array of CSV setting double values containing expected amount of values.

[getCsvDoubles\(String\)](#) - Method in class core.[Settings](#)

Returns an array of CSV setting double values.

[getCsvInts\(String, int\)](#) - Method in class core.[Settings](#)

Returns an array of CSV setting integer values

[getCsvInts\(String\)](#) - Method in class core.[Settings](#)

Returns an array of CSV setting integer values

[getCsvSetting\(String\)](#) - Method in class core.[Settings](#)

Returns a CSV setting.

[getCsvSetting\(String, int\)](#) - Method in class core.[Settings](#)

Returns a CSV setting containing expected amount of values.

[getCurrentMovementModel\(\)](#) - Method in class movement.[ExtendedMovementModel](#)

[getDelta\(\)](#) - Method in class routing.schedule.[ScheduleEntry](#)

[getDestination\(\)](#) - Method in class movement.[EveningTrip](#)

[getDestinationTime\(\)](#) - Method in class routing.schedule.[ScheduleEntry](#)

[getDestMax\(\)](#) - Method in class applications.[PingApplication](#)

[getDestMin\(\)](#) - Method in class applications.[PingApplication](#)

[getDouble\(String, double\)](#) - Method in class core.[ModuleCommunicationBus](#)

Returns a double value from the communication bus.

[getDouble\(\)](#) - Method in class core.[ParetoRNG](#)

Returns a Pareto distributed double value

[getDouble\(String\)](#) - Method in class core.[Settings](#)

Returns a double-valued setting

[getDuration\(\)](#) - Method in class routing.schedule.[ScheduleEntry](#)

Return the time it takes to get from source to destination

[getEncounters\(\)](#) - Method in class report.[TotalEncountersReport](#)

[getEndTime\(\)](#) - Method in class core.[SimScenario](#)

Returns simulation's end time

[getEntries\(\)](#) - Method in class routing.schedule.[ScheduleOracle](#)

Returns all schedule entries

[getEveningActivityControlSystem\(int\)](#) - Static method in class movement.[EveningActivityControlSystem](#)

Returns a reference to a EveningActivityControlSystem with ID provided as parameter.

[getEveningInstructions\(int\)](#) - Method in class movement.[EveningActivityControlSystem](#)

This method gets the instruction for a node, i.e.

[getEventQueues\(\)](#) - Method in class input.[EventQueueHandler](#)

Returns all the loaded event queues

[getEvents\(\)](#) - Static method in class core.[DTN2Manager](#)

Returns the DTN2Events object.

[**getException\(\)**](#) - Method in error core.[SimError](#)

[**getExternalEvents\(\)**](#) - Method in class core.[SimScenario](#)

Returns the (external) event queue(s) of this scenario or null if there aren't any

[**getFreeBufferSize\(\)**](#) - Method in class routing.[MessageRouter](#)

Returns the amount of free space in the buffer.

[**getFrom\(\)**](#) - Method in class core.[Message](#)

Returns the node this message is originally from

[**getFrom\(\)**](#) - Method in class routing.schedule.[ScheduleEntry](#)

[**getFullPropertyName\(String\)**](#) - Method in class core.[Settings](#)

Returns full (namespace prefixed) property name for a setting.

[**getGraphicsPosition\(Coord\)**](#) - Method in class gui.playfield.[PlayField](#)

Returns the graphical presentation location for the given world location

[**getHomeLocation\(\)**](#) - Method in class movement.[HomeActivityMovement](#)

[**getHomeLocation\(\)**](#) - Method in class movement.[WorkingDayMovement](#)

[**getHopCount\(\)**](#) - Method in class core.[Message](#)

Returns the amount of hops this message has passed

[**getHops\(\)**](#) - Method in class core.[Message](#)

Returns a list of nodes this message has passed so far

[**getHost\(\)**](#) - Method in class core.[NetworkInterface](#)

Returns the DTNHost of this interface

[**getHost\(\)**](#) - Method in class routing.[MessageRouter](#)

Returns the host this router is in

[**getHosts\(String\)**](#) - Static method in class core.[DTN2Manager](#)

Returns a collection of DTNHost objects corresponding to the given EID.

[**getHosts\(\)**](#) - Method in class core.[SimScenario](#)

Returns the list of nodes for this scenario.

[**getHosts\(\)**](#) - Method in class core.[World](#)

Returns the hosts in a list

[**getId\(\)**](#) - Method in class core.[Message](#)

Returns the ID of the message

[**getID\(\)**](#) - Method in class input.[MessageEventGenerator](#)

Returns a next free message ID

[**getID\(\)**](#) - Method in class movement.[BusMovement](#)

Returns unique ID of the bus

[**getID\(\)**](#) - Method in class movement.[BusTravellerMovement](#)

[**getID\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**getInfoPanel\(\)**](#) - Method in class gui.[DTNSimGUI](#)

Returns the info panel of the GUI

[**getInitialLocation\(\)**](#) - Method in class movement.[BusMovement](#)

[**getInitialLocation\(\)**](#) - Method in class movement.[BusTravellerMovement](#)

[**getInitialLocation\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**getInitialLocation\(\)**](#) - Method in class movement.[ExternalMovement](#)

[**getInitialLocation\(\)**](#) - Method in class movement.[HomeActivityMovement](#)

[**getInitialLocation\(\)**](#) - Method in class movement.[LinearFormation](#)

Returns the the location of the node in the formation

[**getInitialLocation\(\)**](#) - Method in class movement.[MapBasedMovement](#)

Returns a (random) coordinate that is between two adjacent MapNodes

[getInitialLocation\(\)](#) - Method in class movement.[MapRouteMovement](#)

Returns the first stop on the route

[getInitialLocation\(\)](#) - Method in class movement.[MovementModel](#)

Returns a new initial placement for a node

[getInitialLocation\(\)](#) - Method in class movement.[OfficeActivityMovement](#)

[getInitialLocation\(\)](#) - Method in class movement.[RandomWalk](#)

Returns a possible (random) placement for a host

[getInitialLocation\(\)](#) - Method in class movement.[RandomWaypoint](#)

Returns a possible (random) placement for a host

[getInitialLocation\(\)](#) - Method in class movement.[StationaryMovement](#)

Returns the only location of this movement model

[getInitialLocation\(\)](#) - Method in class movement.[WorkingDayMovement](#)

[getInstance\(\)](#) - Static method in class core.[SimClock](#)

Get the instance of the class that can also change the time.

[getInstance\(\)](#) - Static method in class core.[SimScenario](#)

Returns the SimScenario instance and creates one if it doesn't exist yet

[getInt\(String, int\)](#) - Method in class core.[ModuleCommunicationBus](#)

Returns an integer value from the communication bus.

[getInt\(String\)](#) - Method in class core.[Settings](#)

Returns an integer-valued setting

[getIntAverage\(List<Integer>\)](#) - Method in class report.[Report](#)

Returns the average of integer values stored in a List

[getInterface\(int\)](#) - Method in class core.[DTNHost](#)

Find the network interface based on the index

[getInterface\(String\)](#) - Method in class core.[DTNHost](#)

Find the network interface based on the interfacetype

[getInterfaces\(\)](#) - Method in class core.[DTNHost](#)

Returns the interface objects of the node

[getInterfaces\(\)](#) - Method in class interfaces.[ConnectivityGrid.GridCell](#)

Returns a list of of interfaces in this cell

[getInterfaceType\(\)](#) - Method in class core.[NetworkInterface](#)

For checking what interface type this interface is

[getInterval\(\)](#) - Method in class applications.[PingApplication](#)

[getIntMedian\(List<Integer>\)](#) - Method in class report.[Report](#)

Returns the median of integer values stored in a List

[getIntTime\(\)](#) - Static method in class core.[SimClock](#)

Returns the current time rounded to the nearest integer

[getKey\(\)](#) - Method in class core.[Tuple](#)

Returns the key

[getLastLocation\(\)](#) - Method in class movement.[BusTravellerMovement](#)

[getLastLocation\(\)](#) - Method in class movement.[EveningActivityMovement](#)

[getLastLocation\(\)](#) - Method in class movement.[HomeActivityMovement](#)

[getLastLocation\(\)](#) - Method in class movement.[MapBasedMovement](#)

[getLastLocation\(\)](#) - Method in class movement.[MapRouteMovement](#)

[getLastLocation\(\)](#) - Method in class movement.[OfficeActivityMovement](#)

[getLastLocation\(\)](#) - Method in class movement.[RandomWalk](#)

[**getLastLocation\(\)**](#) - Method in interface movement.[SwitchableMovement](#)

Get the last location the getPath() of this movement model has returned

[**getLastPing\(\)**](#) - Method in class applications.[PingApplication](#)

[**getLastTimeStamp\(\)**](#) - Method in class input.[ExternalMovementReader](#)

Returns the time stamp where the last moves read with [ExternalMovementReader.readNextMovements\(\)](#) belong to.

[**getLastUpdateTime\(\)**](#) - Method in class routing.maxprop.[MeetingProbabilitySet](#)

Returns the time when this probability set was last updated

[**getLocation\(\)**](#) - Method in class core.[DTNHost](#)

Returns the current location of this host.

[**getLocation\(\)**](#) - Method in class core.[NetworkInterface](#)

Returns the current location of the host of this interface.

[**getLocation\(\)**](#) - Method in class movement.[BusTravellerMovement](#)

Get the location where the bus is located when it has moved its path

[**getLocation\(\)**](#) - Method in class movement.[EveningTrip](#)

[**getLocation\(\)**](#) - Method in class movement.map.[MapNode](#)

Returns the location of the node

[**getMap\(\)**](#) - Method in class core.[SimScenario](#)

Returns the SimMap this scenario uses, or null if scenario doesn't use any map

[**getMap\(\)**](#) - Method in class input.[WKTMapReader](#)

Returns new a SimMap that is based on the read map

[**getMap\(\)**](#) - Method in class movement.[BusControlSystem](#)

Get the underlying map of the system

[**getMap\(\)**](#) - Method in class movement.[MapBasedMovement](#)

Returns the SimMap this movement model uses

[**getMaxBound\(\)**](#) - Method in class movement.map.[SimMap](#)

Returns the lower right corner coordinate of the map

[**getMaxGroupSize\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**getMaxHostRange\(\)**](#) - Method in class core.[SimScenario](#)

Returns how long range the hosts' radios have

[**getMaxTime\(\)**](#) - Method in class input.[ExternalMovementReader](#)

Returns offset maxTime

[**getMaxX\(\)**](#) - Method in class input.[ExternalMovementReader](#)

Returns offset maxX

[**getMaxX\(\)**](#) - Method in class movement.[ClusterMovement](#)

[**getMaxX\(\)**](#) - Method in class movement.[ExternalMovement](#)

[**getMaxX\(\)**](#) - Method in class movement.[MovementModel](#)

Returns the largest X coordinate value this model uses

[**getMaxY\(\)**](#) - Method in class input.[ExternalMovementReader](#)

Returns offset maxY

[**getMaxY\(\)**](#) - Method in class movement.[ClusterMovement](#)

[**getMaxY\(\)**](#) - Method in class movement.[ExternalMovement](#)

[**getMaxY\(\)**](#) - Method in class movement.[MovementModel](#)

Returns the largest Y coordinate value this model uses

[**getMedian\(List<Double>\)**](#) - Method in class report.[Report](#)

Returns the median of double values stored in a List

[**getMeetingSpotForID\(int\)**](#) - Method in class movement.[EveningActivityControlSystem](#)

Get the meeting spot for the node

[**getMessage\(\)**](#) - Method in class core.[Connection](#)

Gets the message that this connection is currently transferring.

[**getMessage\(String\)**](#) - Method in class routing.[MessageRouter](#)

Returns a message by ID.

[**getMessageCollection\(\)**](#) - Method in class core.[DTNHost](#)

Returns the messages in a collection.

[**getMessageCollection\(\)**](#) - Method in class routing.[MessageRouter](#)

Returns a reference to the messages of this router in collection.

[**getMessagesForConnected\(\)**](#) - Method in class routing.[ActiveRouter](#)

Returns a list of message-connections tuples of the messages whose recipient is some host that we're connected to at the moment.

[**getMessagesWithCopiesLeft\(\)**](#) - Method in class routing.[SprayAndWaitRouter](#)

Creates and returns a list of messages this router is currently carrying and still has copies left to distribute (nrof copies > 1).

[**getMinBound\(\)**](#) - Method in class movement.map.[SimMap](#)

Returns the upper left corner coordinate of the map

[**getMinGroupSize\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**getMinTime\(\)**](#) - Method in class input.[ExternalMovementReader](#)

Returns offset minTime

[**getMinX\(\)**](#) - Method in class input.[ExternalMovementReader](#)

Returns offset minX

[**getMinY\(\)**](#) - Method in class input.[ExternalMovementReader](#)

Returns offset minY

[**getMoreInfo\(\)**](#) - Method in class routing.[RoutingInfo](#)

Returns the child routing infos of this info.

[**getName\(\)**](#) - Method in class core.[SimScenario](#)

Returns the name of the simulation run

[**getNameSpace\(\)**](#) - Method in class core.[Settings](#)

Returns the namespace of the settings object

[**getNearInterfaces\(NetworkInterface\)**](#) - Method in class interfaces.[ConnectivityGrid](#)

Returns all interfaces using the same technology and channel that are in neighboring cells

[**getNearInterfaces\(NetworkInterface\)**](#) - Method in class interfaces.[ConnectivityOptimizer](#)

Finds all network interfaces that might be located so that they can be connected with the network interface

[**getNeighbors\(\)**](#) - Method in class movement.map.[MapNode](#)

Returns the neighbors of this node.

[**getNextWaypoint\(\)**](#) - Method in class movement.[Path](#)

Returns the next waypoint on this path

[**getNodeByAddress\(int\)**](#) - Method in class core.[World](#)

Returns a node from the world by its address

[**getNodeByCoord\(Coord\)**](#) - Method in class movement.map.[SimMap](#)

Returns a MapNode at given coordinates or null if there's no MapNode in the location of the coordinate

[**getNodeRelationships\(\)**](#) - Method in class report.[UniqueEncountersReport](#)

[**getNodes\(\)**](#) - Method in class input.[WKTMapReader](#)

Returns the map nodes that were read in a collection

[**getNodes\(\)**](#) - Method in class movement.map.[SimMap](#)

Returns all the map nodes in a list

[**getNodesHash\(\)**](#) - Method in class input.[WKTMapReader](#)

Returns the original Map object that was used to read the map

[**getNrofMessages\(\)**](#) - Method in class core.[DTNHost](#)

Returns the number of messages this node is carrying.

[**getNrofMessages\(\)**](#) - Method in class routing.[MessageRouter](#)

Returns the number of messages this router has

[**getNrofStops\(\)**](#) - Method in class movement.map.[MapRoute](#)

Returns the number of stops on this route

[**getOfficeLocation\(\)**](#) - Method in class movement.[OfficeActivityMovement](#)

[**getOfficeLocation\(\)**](#) - Method in class movement.[WorkingDayMovement](#)

[**getOffset\(\)**](#) - Method in class movement.map.[SimMap](#)

Returns the offset that has been caused by translates made to this map (does NOT take into account mirroring).

[**getOkMapNodeTypes\(\)**](#) - Method in class movement.[MapBasedMovement](#)

Returns map node types that are OK for this movement model in an array or null if all values are considered ok

[**getOldestMessage\(boolean\)**](#) - Method in class routing.[ActiveRouter](#)

Returns the oldest (by receive time) message in the message buffer (that is not being sent if excludeMsgBeingSent is true).

[**getOldestMessage\(boolean\)**](#) - Method in class routing.[MaxPropRouter](#)

Returns the next message that should be dropped, according to MaxProp's message ordering scheme (see MaxPropTupleComparator).

[**getOldestMessage\(boolean\)**](#) - Method in class routing.[MaxPropRouterWithEstimation](#)

Returns the next message that should be dropped, according to MaxProp's message ordering scheme (see MaxPropTupleComparator).

[**getOtherInterface\(NetworkInterface\)**](#) - Method in class core.[Connection](#)

Returns the interface in the other end of the connection

[**getOtherNode\(DTNHost\)**](#) - Method in class core.[Connection](#)

Returns the node in the other end of the connection

[**getParentFrame\(\)**](#) - Method in class gui.[DTNSimGUI](#)

Returns the parent frame (window) of the gui.

[**getParser\(DTNHost\)**](#) - Static method in class core.[DTN2Manager](#)

Returns the ECL parser associated with the host.

[**getParserHandler\(int, String, int\)**](#) - Method in class input.[DTN2Events](#)

Creates a parser handler for the given host.

[**getPath\(\)**](#) - Method in class core.[DTNHost](#)

Returns the Path this node is currently traveling or null if no path is in use at the moment.

[**getPath\(\)**](#) - Method in class movement.[BusMovement](#)

[**getPath\(\)**](#) - Method in class movement.[BusTravellerMovement](#)

[**getPath\(\)**](#) - Method in class movement.[CarMovement](#)

[**getPath\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**getPath\(\)**](#) - Method in class movement.[EveningTrip](#)

[**getPath\(\)**](#) - Method in class movement.[ExtendedMovementModel](#)

[**getPath\(\)**](#) - Method in class movement.[ExternalMovement](#)

[**getPath\(\)**](#) - Method in class movement.[HomeActivityMovement](#)

[**getPath\(\)**](#) - Method in class movement.[LinearFormation](#)

Returns a single coordinate path (using the only possible coordinate)

[**getPath\(\)**](#) - Method in class movement.[MapBasedMovement](#)

[**getPath\(\)**](#) - Method in class movement.[MapRouteMovement](#)

[**getPath\(\)**](#) - Method in class movement.[MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).

[**getPath\(\)**](#) - Method in class movement.[OfficeActivityMovement](#)

[**getPath\(\)**](#) - Method in class movement.[RandomWalk](#)

[**getPath\(\)**](#) - Method in class movement.RandomWaypoint

[**getPath\(\)**](#) - Method in class movement.ShortestPathMapBasedMovement

[**getPath\(\)**](#) - Method in class movement.StationaryMovement

Returns a single coordinate path (using the only possible coordinate)

[**getPingSize\(\)**](#) - Method in class applications.PingApplication

[**getPlayFieldScroll\(\)**](#) - Method in class gui.MainWindow

Returns a reference of the play field scroll panel

[**getPongSize\(\)**](#) - Method in class applications.PingApplication

[**getPredFor\(DTNHost\)**](#) - Method in class routing.ProphetRouter

Returns the current prediction (P) value for a host or 0 if entry for the host doesn't exist.

[**getPredFor\(DTNHost\)**](#) - Method in class routing.ProphetRouterWithEstimation

Returns the current prediction (P) value for a host or 0 if entry for the host doesn't exist.

[**getProbFor\(Integer\)**](#) - Method in class routing.maxprop.MeetingProbabilitySet

Returns the current delivery probability value for the given node index

[**getProperty\(String\)**](#) - Method in class core.Message

Returns an object that was stored to this message using the given key.

[**getProperty\(String\)**](#) - Method in class core.ModuleCommunicationBus

Returns an object that was stored using the given key.

[**getRandomCoorinateInsideOffice\(\)**](#) - Method in class movement.OfficeActivityMovement

[**getReceiveTime\(\)**](#) - Method in class core.Message

Returns the time when this message was received

[**getRemainingByteCount\(\)**](#) - Method in class core.CBRCConnection

Returns the amount of bytes to be transferred before ongoing transfer is ready or 0 if there's no ongoing transfer or it has finished already

[**getRemainingByteCount\(\)**](#) - Method in class core.Connection

Returns the amount of bytes to be transferred before ongoing transfer is ready or 0 if there's no ongoing transfer or it has finished already

[**getRemainingByteCount\(\)**](#) - Method in class core.VBRCConnection

Returns the amount of bytes to be transferred before ongoing transfer is ready or 0 if there's no ongoing transfer or it has finished already

[**getReporter\(\)**](#) - Static method in class core.DTN2Manager

Returns reference to the DTN2Reporter object.

[**getRequest\(\)**](#) - Method in class core.Message

Returns the message this message is response to or null if this is not a response message

[**getResponseSize\(\)**](#) - Method in class core.Message

Returns the size of the requested response message or 0 if no response is requested.

[**getRouter\(\)**](#) - Method in class core.DTNHost

Returns the router of this host

[**getRoutingInfo\(\)**](#) - Method in class core.DTNHost

Returns routing info of this host's router.

[**getRoutingInfo\(\)**](#) - Method in class routing.MaxPropRouter

[**getRoutingInfo\(\)**](#) - Method in class routing.MaxPropRouterWithEstimation

[**getRoutingInfo\(\)**](#) - Method in class routing.MessageRouter

Returns routing information about this router.

[**getRoutingInfo\(\)**](#) - Method in class routing.ProphetRouter

[**getRoutingInfo\(\)**](#) - Method in class routing.ProphetRouterWithEstimation

[**getScale\(\)**](#) - Static method in class gui.playfield.PlayFieldGraphic

Returns the currently used scaling factor

[**getScenarioName\(\)**](#) - Method in class report.[Report](#)

Returns the name of the scenario as read from the settings

[**getSeed\(\)**](#) - Method in class applications.[PingApplication](#)

[**getSetting\(String\)**](#) - Method in class core.[Settings](#)

Returns a String-valued setting.

[**getSettings\(\)**](#) - Method in class report.[Report](#)

Returns a Settings object initialized for the report class' name space that uses "[Report](#)" as the secondary name space.

[**getShoppingLocation\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**getShoppingLocation\(\)**](#) - Method in class movement.[WorkingDayMovement](#)

[**getShoppingLocationAndGetReady\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

Sets the node ready to start a shopping trip.

[**getShortestPath\(MapNode, MapNode\)**](#) - Method in class movement.map.[DijkstraPathFinder](#)

Finds and returns a shortest path between two map nodes

[**getShortestPath\(Integer, Integer, double\)**](#) - Method in class routing.schedule.[ScheduleDijkstra](#)

Finds and returns the fastest path between two destinations

[**getSimTime\(\)**](#) - Method in class report.[Report](#)

Returns the current simulation time from the SimClock

[**getSize\(\)**](#) - Method in class core.[Message](#)

Returns the size of the message (in bytes)

[**getSizeX\(\)**](#) - Method in class core.[World](#)

Returns the x-size (width) of the world

[**getSizeY\(\)**](#) - Method in class core.[World](#)

Returns the y-size (height) of the world

[**getSpeed\(\)**](#) - Method in class core.[CBRConnection](#)

returns the current speed of the connection

[**getSpeed\(\)**](#) - Method in class core.[Connection](#)

Gets the current connection speed

[**getSpeed\(\)**](#) - Method in class core.[VBRConnection](#)

returns the current speed of the connection

[**getSpeed\(\)**](#) - Method in class movement.[Path](#)

Returns the speed towards the next waypoint (asked with [Path.getNextWaypoint\(\)](#)).

[**getSpeeds\(\)**](#) - Method in class movement.[Path](#)

[**getState\(\)**](#) - Method in class movement.[BusTravellerMovement](#)

[**getStops\(\)**](#) - Method in class movement.map.[MapRoute](#)

[**getStops\(\)**](#) - Method in class movement.[MapRouteMovement](#)

Returns the list of stops on the route

[**getTime\(\)**](#) - Static method in class core.[SimClock](#)

Returns the current time (seconds since start)

[**getTime\(\)**](#) - Method in class input.[ExternalEvent](#)

Returns the time when this event should happen.

[**getTime\(\)**](#) - Method in class routing.schedule.[ScheduleEntry](#)

Returns time + delta

[**getTo\(\)**](#) - Method in class core.[Message](#)

Returns the node this message is originally to

[**getTo\(\)**](#) - Method in class routing.schedule.[ScheduleEntry](#)

[**getTotalBytesTransferred\(\)**](#) - Method in class core.[Connection](#)

Returns the total amount of bytes this connection has transferred so far (including all transfers).

[**getTransferDoneTime\(\)**](#) - Method in class core.[CBRConnection](#)

Gets the transferdonetime

[**getTransmitRange\(\)**](#) - Method in class core.[NetworkInterface](#)

Returns the transmit range of this network layer

[**getTransmitSpeed\(\)**](#) - Method in class core.[NetworkInterface](#)

Returns the transmit speed of this network layer

[**getTransmitSpeed\(\)**](#) - Method in class interfaces.[InterferenceLimitedInterface](#)

Returns the transmit speed of this network layer

[**getTtl\(\)**](#) - Method in class core.[Message](#)

Returns the time to live (minutes) of the message or Integer.MAX_VALUE if the TTL is infinite.

[**getUniqueId\(\)**](#) - Method in class core.[Message](#)

Returns an ID that is unique per message instance (different for replicates too)

[**getUpdateInterval\(\)**](#) - Method in class core.[SimScenario](#)

Returns update interval (simulated seconds) of the simulation

[**getUpdateInterval\(\)**](#) - Method in class gui.[GUIControls](#)

Returns the selected update interval of GUI

[**getUpdateListeners\(\)**](#) - Method in class core.[SimScenario](#)

Returns the list of registered update listeners

[**getUsageCount\(\)**](#) - Method in class routing.schedule.[ScheduleEntry](#)

[**getValue\(\)**](#) - Method in class core.[Tuple](#)

Returns the value

[**getVariance\(List<Double>\)**](#) - Method in class report.[Report](#)

Returns the variance of the values in the List.

[**getVia\(\)**](#) - Method in class routing.schedule.[ScheduleEntry](#)

[**getWaitTimeAtEnd\(\)**](#) - Method in class movement.[EveningTrip](#)

[**getWorld\(\)**](#) - Method in class core.[SimScenario](#)

Returns the World object of this scenario

[**getWorldPosition\(Coord\)**](#) - Method in class gui.playfield.[PlayField](#)

Returns a world location for a given graphical location.

[**getWorldSizeX\(\)**](#) - Method in class core.[SimScenario](#)

Returns the width of the world

[**getWorldSizeY\(\)**](#) - Method in class core.[SimScenario](#)

Returns the height of the world

[**getX\(\)**](#) - Method in class core.[Coord](#)

Returns the x coordinate

[**getY\(\)**](#) - Method in class core.[Coord](#)

Returns the y coordinate

[**gotEvent\(String, Object, Application, DTNHost\)**](#) - Method in interface core.[ApplicationListener](#)

Application has generated an event.

[**gotEvent\(String, Object, Application, DTNHost\)**](#) - Method in class report.[PingAppReporter](#)

[**GRANULARITY**](#) - Static variable in class report.[ContactTimesReport](#)

Granularity -setting id ("granularity").

[**granularity**](#) - Variable in class report.[ContactTimesReport](#)

How many seconds are grouped in one group

[**GRANULARITY**](#) - Static variable in class report.[EnergyLevelReport](#)

Reporting granularity -setting id ("granularity").

[**granularity**](#) - Variable in class report.[EnergyLevelReport](#)

value of the granularity setting

[**GRANULARITY**](#) - Static variable in class report.[MessageLocationReport](#)

Reporting granularity -setting id ("granularity").

[**granularity**](#) - Variable in class report.[MessageLocationReport](#)

value of the granularity setting

[**GRAPH_NAME**](#) - Static variable in class report.[AdjacencyGraphvizReport](#)

Name of the graphviz report ("adjgraph")

[**GRAPH_NAME**](#) - Static variable in class report.[MessageGraphvizReport](#)

Name of the graphviz report ("msggraph")

GROUP_ID_S - Static variable in class core.[SimScenario](#)

group id -setting id ("groupID")

GROUP_NS - Static variable in class core.[SimScenario](#)

namespace for host group settings ("Group")

gui - package gui

Contains the classes of Graphical User Interface.

gui.playfield - package gui.playfield

Contains the classes of Graphical User Interface's playfield -view (the graphical presentation of the nodes' locations and other information).

GUIControls - Class in [gui](#)

GUI's control panel

GUIControls(DTNSimGUI, PlayField) - Constructor for class [gui.GUIControls](#)

H

handle(Message, DTNHost) - Method in class applications.[PingApplication](#)

Handles an incoming message.

handle(Message, DTNHost) - Method in class core.[Application](#)

This method handles application functionality related to processing of the bundle.

hashCode() - Method in class core.[Coord](#)

Returns a hash code for this coordinate (actually a hash of the String made of the coordinates)

hashCode() - Method in class report.[ContactTimesReport.ConnectionInfo](#)

Returns the same hash for ConnectionInfos that have the same two hosts.

hasMessage(String) - Method in class routing.[MessageRouter](#)

Checks if this router has a message with certain id buffered.

hasNext() - Method in class movement.[Path](#)

Returns true if the path has more waypoints, false if not

HEADER - Static variable in class report.[CreatedMessagesReport](#)

HEADER - Static variable in class report.[DeliveredMessagesReport](#)

HEADER - Static variable in class report.[MessageDelayReport](#)

HEADER - Static variable in class report.[MessageDeliveryReport](#)

HEADER - Static variable in class report.[MessageReport](#)

HEADER - Static variable in class report.[TotalContactTimeReport](#)

The header of every report file

HOME_LOCATIONS_FILE_SETTING - Static variable in class movement.[HomeActivityMovement](#)

HomeActivityMovement - Class in [movement](#)

A Class to model movement at home.

HomeActivityMovement(Settings) - Constructor for class movement.[HomeActivityMovement](#)

Creates a new instance of HomeActivityMovement

HomeActivityMovement(HomeActivityMovement) - Constructor for class movement.[HomeActivityMovement](#)

Creates a new instance of HomeActivityMovement from a prototype

host - Variable in class core.[DTN2Manager.EIDHost](#)

host - Variable in class core.[NetworkInterface](#)

host_id - Variable in class core.[DTN2Manager.EIDHost](#)

HOST_RANGE_S - Static variable in class input.[MessageEventGenerator](#)

Sender/receiver address range -setting id ("hosts").

[**hostRange**](#) - Variable in class input.[MessageEventGenerator](#)

Range of host addresses that can be senders or receivers

[**hosts**](#) - Variable in class core.[SimScenario](#)

List of hosts in this simulation

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in interface core.[ConnectionListener](#)

Method is called when two hosts are connected.

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class gui.[EventLogPanel](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[AdjacencyGraphvizReport](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ConnectivityDtnsim2Report](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ConnectivityONEReport](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ContactsDuringAnICTReport](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ContactsPerHourReport](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ContactTimesReport](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[EncountersVSUniqueEncountersReport](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[EventLogReport](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[InterContactTimesReport](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[TotalEncountersReport](#)

[**hostsConnected\(DTNHost, DTNHost\)**](#) - Method in class report.[UniqueEncountersReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in interface core.[ConnectionListener](#)

Method is called when connection between hosts is disconnected.

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class gui.[EventLogPanel](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[AdjacencyGraphvizReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ConnectivityDtnsim2Report](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ConnectivityONEReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ContactsDuringAnICTReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ContactsPerHourReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[ContactTimesReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[EncountersVSUniqueEncountersReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[EventLogReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[InterContactTimesReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[TotalContactTimeReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[TotalEncountersReport](#)

[**hostsDisconnected\(DTNHost, DTNHost\)**](#) - Method in class report.[UniqueEncountersReport](#)

I

[**id**](#) - Variable in class input.[MessageEvent](#)

identifier of the message

[**idPrefix**](#) - Variable in class input.[MessageEventGenerator](#)

Prefix for the messages

[**incomingBundle\(String, CLAParser.BundleAttributes\)**](#) - Method in class input.[DTN2Events.ParserHandler](#)

[**increaseTimeCount\(double\)**](#) - Method in class report.[ContactTimesReport](#)

Increases the amount of times a certain time value has been seen.

[**increaseUsageCount\(\)**](#) - Method in class routing.schedule.[ScheduleEntry](#)

[**INFINITE_SET_SIZE**](#) - Static variable in class routing.maxprop.[MeetingProbabilitySet](#)

[**INFINITE_TTL**](#) - Static variable in class core.[Message](#)

Value for infinite TTL of message

[**InfoPanel**](#) - Class in [gui](#)

Information panel that shows data of selected messages and nodes.

[**InfoPanel\(DTNSimGUI\)**](#) - Constructor for class [gui](#).[InfoPanel](#)

[**init\(String\)**](#) - Static method in class core.[Settings](#)

Initializes the settings all Settings objects will use.

[**init\(Reader\)**](#) - Method in class input.[WKTReader](#)

Initialize the reader to use a certain input reader

[**init\(\)**](#) - Method in class report.[AdjacencyGraphvizReport](#)

[**init\(\)**](#) - Method in class report.[ContactsDuringAnICTReport](#)

[**init\(\)**](#) - Method in class report.[ContactsPerHourReport](#)

[**init\(\)**](#) - Method in class report.[ContactTimesReport](#)

[**init\(\)**](#) - Method in class report.[CreatedMessagesReport](#)

[**init\(\)**](#) - Method in class report.[DeliveredMessagesReport](#)

[**init\(\)**](#) - Method in class report.[DistanceDelayReport](#)

[**init\(\)**](#) - Method in class report.[MessageDelayReport](#)

[**init\(\)**](#) - Method in class report.[MessageDeliveryReport](#)

[**init\(\)**](#) - Method in class report.[MessageGraphvizReport](#)

[**init\(\)**](#) - Method in class report.[MessageReport](#)

[**init\(\)**](#) - Method in class report.[MessageStatsReport](#)

[**init\(\)**](#) - Method in class report.[Report](#)

Initializes the report output.

[**init\(\)**](#) - Method in class report.[TotalContactTimeReport](#)

[**init\(DTNHost, List<MessageListener>\)**](#) - Method in class routing.[ActiveRouter](#)

[**init\(DTNHost, List<MessageListener>\)**](#) - Method in class routing.[MessageRouter](#)

Initializes the router; i.e.

[**INIT ENERGY_S**](#) - Static variable in class routing.[EnergyAwareRouter](#)

Initial units of energy -setting id ("intialEnergy").

[**INITIAL SPEED SELECTION**](#) - Static variable in class gui.[GUIControls](#)

index of initial update speed setting

[**initialLocation\(DTNHost, Coord\)**](#) - Method in interface core.[MovementListener](#)

Method is called when a host receives its initial location from movement model.

[**initialLocation\(DTNHost, Coord\)**](#) - Method in class report.[MovementNs2Report](#)

[**initialNrofCopies**](#) - Variable in class routing.[SprayAndWaitRouter](#)

[**input**](#) - package input

Provides interfaces and classes for reading input data from external sources.

[**InterContactTimesReport**](#) - Class in [report](#)

Reports the inter-contact time (i.e., the time between the end of previous contact and the beginning of a new contact between two hosts) distribution.

[**InterContactTimesReport\(\)**](#) - Constructor for class report.[InterContactTimesReport](#)

[**interfaceId**](#) - Variable in class input.[ConnectionEvent](#)

What is the interface number for this event

[**INTERFACENAME_S**](#) - Static variable in class core.[SimScenario](#)

interface name in the group -setting id ("interface")

[**interfaces**](#) - package interfaces

[**interfacetype**](#) - Variable in class core.[NetworkInterface](#)

[**InterferenceLimitedInterface**](#) - Class in [interfaces](#)

A simple Network Interface that provides a variable bit-rate service, where the bit-rate depends on the number of other transmitting stations within range. The current transmit speed is updated only if there are ongoing transmissions.

[**InterferenceLimitedInterface\(Settings\)**](#) - Constructor for class interfaces.[InterferenceLimitedInterface](#)

[**InterferenceLimitedInterface\(InterferenceLimitedInterface\)**](#) - Constructor for class interfaces.[InterferenceLimitedInterface](#)

Copy constructor

[**INTERVAL_SETTING**](#) - Static variable in class report.[Report](#)

The interval (simulated seconds) of creating new settings files -setting id ("interval")

[**INTERVALLED_FORMAT**](#) - Static variable in class report.[Report](#)

Suffix for reports that are created on n second intervals

[**INTNAME_S**](#) - Static variable in class core.[SimScenario](#)

interface name -setting id ("name")

[**INTTYPE_NS**](#) - Static variable in class core.[SimScenario](#)

namespace for interface type settings ("Interface")

[**INTTYPE_S**](#) - Static variable in class core.[SimScenario](#)

interface type -setting id ("type")

[**invScale\(double\)**](#) - Static method in class gui.playfield.[PlayFieldGraphic](#)

Performs an inverse of the scaling procedure with current scale.

[**isActive\(\)**](#) - Method in class core.[DTNHost](#)

Returns true if this node is active (false if not)

[**isActive\(\)**](#) - Method in class movement.[ActivenessHandler](#)

Returns true if node should be active at the moment

[**isActive\(\)**](#) - Method in class movement.[ExternalMovement](#)

[**isActive\(\)**](#) - Method in class movement.[MovementModel](#)

Returns true if this node is active at the moment (false if not)

[**isBinary**](#) - Variable in class routing.[SprayAndWaitRouter](#)

[**isBinaryEeFile\(File\)**](#) - Static method in class input.[BinaryEventsReader](#)

Checks if the given file is a binary external events file

[**isConnected\(NetworkInterface\)**](#) - Method in class core.[NetworkInterface](#)

Returns true if the given NetworkInterface is connected to this host.

[**isDeliveredMessage\(Message\)**](#) - Method in class routing.[MessageRouter](#)

Returns true if a full message with same ID as the given message has been received by this host as the **final** recipient (at least once).

[**isDone\(\)**](#) - Method in class input.[WKTReader](#)

Returns true if the whole file has been read

[**isFfw\(\)**](#) - Method in class gui.[GUIControls](#)

Is fast forward turned on

[**isFull\(\)**](#) - Method in class movement.[EveningTrip](#)

[**isIncomingMessage\(String\)**](#) - Method in class routing.[MessageRouter](#)

Returns true if a message with the given ID is one of the currently incoming messages, false if not

[**isInitiator\(DTNHost\)**](#) - Method in class core.[Connection](#)

Returns true if the given node is the initiator of the connection, false otherwise

[**isKnownType\(String\)**](#) - Method in class input.[WKTReader](#)

Returns true if type is one of the known WKT types

[**isMessageTransferred\(\)**](#) - Method in class core.[CBRConnection](#)

Returns true if the current message transfer is done.

[**isMessageTransferred\(\)**](#) - Method in class core.[Connection](#)

Returns true if the current message transfer is done

[**isMessageTransferred\(\)**](#) - Method in class core.[VBRConnection](#)

Returns true if the current message transfer is done.

[**isMirrored\(\)**](#) - Method in class movement.map.[SimMap](#)

Returns true if this map has been mirrored after reading

[**isPassive\(\)**](#) - Method in class applications.[PingApplication](#)

[**isPaused\(\)**](#) - Method in class gui.[GUIControls](#)

Has user requested the simulation to be paused

[**isReady\(\)**](#) - Method in class movement.[BusTravellerMovement](#)

[**isReady\(\)**](#) - Method in class movement.[CarMovement](#)

[**isReady\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**isReady\(\)**](#) - Method in class movement.[HomeActivityMovement](#)

[**isReady\(\)**](#) - Method in class movement.[MapBasedMovement](#)

[**isReady\(\)**](#) - Method in class movement.[OfficeActivityMovement](#)

[**isReady\(\)**](#) - Method in class movement.[RandomWalk](#)

[**isReady\(\)**](#) - Method in interface movement.[SwitchableMovement](#)

Checks if the movement model is finished doing its task and it's time to switch to the next movement model.

[**isReadyForTransfer\(\)**](#) - Method in class core.[Connection](#)

Returns true if the connection is ready to transfer a message (connection is up and there is no message being transferred).

[**isReadyToShop\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

Checks if a node is at the correct place where the shopping begins

[**isResponse\(\)**](#) - Method in class core.[Message](#)

Returns true if this message is a response message

[**isScanning\(\)**](#) - Method in class core.[NetworkInterface](#)

Checks if this interface is currently in the scanning mode

isSending(String) - Method in class routing.[ActiveRouter](#)

Returns true if this router is currently sending a message with msgId.

isTransferring() - Method in class interfaces.[InterferenceLimitedInterface](#)

Returns true if this interface is actually transmitting data

isTransferring() - Method in class routing.[ActiveRouter](#)

Returns true if this router is transferring something at the moment or some transfer has not been finalized.

isType(int) - Method in class movement.map.[MapNode](#)

Returns true if this node is of given type, false if none of node's type(s) match to given type or node doesn't have type at all

isType(int[]) - Method in class movement.map.[MapNode](#)

Returns true if the node's types match any of the given types

isUp() - Method in class core.[Connection](#)

Returns true if the connection is up

isUp - Variable in class input.[ConnectionEvent](#)

Is this a "connection up" event

isWarmup() - Method in class report.[Report](#)

Returns true if the warm up period is still ongoing (simTime < warmup)

isWarmupID(String) - Method in class report.[Report](#)

Returns true if the given ID is in the warm up ID set

isWithinRange(NetworkInterface) - Method in class core.[NetworkInterface](#)

Returns true if another interface is within radio range of this interface and this interface is also within radio range of the another interface.

L

lastMapNode - Variable in class movement.[MapBasedMovement](#)

node where the last path ended or node next to initial placement

lastUpdate - Variable in class report.[EnergyLevelReport](#)

time of last update

lastUpdate - Variable in class report.[MessageLocationReport](#)

time of last update

lastUpdate - Variable in class ui.[DTNSimUI](#)

simtime of last UI update

LINEAR FORMATION_NS - Static variable in class movement.[LinearFormation](#)

Name space of the settings (append to group name space)

LinearFormation - Class in [movement](#)

A stationary "movement" model where nodes do not move but are in linear formation (i.e., in a line).

LinearFormation(Settings) - Constructor for class movement.[LinearFormation](#)

Creates a new movement model based on a Settings object's settings.

LinearFormation(LinearFormation) - Constructor for class movement.[LinearFormation](#)

Copy constructor.

LINESTRING - Static variable in class input.[WKTReader](#)

known WKT type LINESTRING

LOCATION_S - Static variable in class movement.[StationaryMovement](#)

Per node group setting for setting the location ("nodeLocation")

LOG UP INTERVAL - Static variable in class gui.[EventLogPanel](#)

How often the log is updated (milliseconds)

M

main(String[]) - Static method in class core.[DTNSim](#)

Starts the user interface with given arguments.

MainWindow - Class in [gui](#)

Main window for the program.

[**MainWindow\(String, World, PlayField, GUIControls, InfoPanel, EventLogPanel, DTNSimGUI\)**](#) - Constructor for class gui.[MainWindow](#)

[**makeRoomForMessage\(int\)**](#) - Method in class routing.[ActiveRouter](#)

Removes messages from the buffer (oldest first) until there's enough space for the new message.

[**makeRoomForNewMessage\(int\)**](#) - Method in class routing.[ActiveRouter](#)

Tries to make room for a new message.

[**MAP_BASE_MOVEMENT_NS**](#) - Static variable in class movement.[MapBasedMovement](#)

map based movement model's settings namespace ("MapBasedMovement")

[**MAP_SELECT_S**](#) - Static variable in class movement.[MapBasedMovement](#)

Per node group setting for selecting map node types that are OK for this node group to traverse through.

[**MapBasedMovement**](#) - Class in [movement](#)

Map based movement model which gives out Paths that use the roads of a SimMap.

[**MapBasedMovement\(Settings\)**](#) - Constructor for class movement.[MapBasedMovement](#)

Creates a new MapBasedMovement based on a Settings object's settings.

[**MapBasedMovement\(Settings, SimMap, int\)**](#) - Constructor for class movement.[MapBasedMovement](#)

Creates a new MapBasedMovement based on a Settings object's settings but with different SimMap

[**MapBasedMovement\(MapBasedMovement\)**](#) - Constructor for class movement.[MapBasedMovement](#)

Copyconstructor.

[**MapGraphic**](#) - Class in [gui.playfield](#)

PlayfieldGraphic for SimMap visualization

[**MapGraphic\(SimMap\)**](#) - Constructor for class gui.playfield.[MapGraphic](#)

[**MapNode**](#) - Class in [movement.map](#)

A node in a SimMap.

[**MapNode\(Coord\)**](#) - Constructor for class movement.map.[MapNode](#)

Constructor.

[**MapRoute**](#) - Class in [movement.map](#)

A route that consists of map nodes.

[**MapRoute\(int, List<MapNode>\)**](#) - Constructor for class movement.map.[MapRoute](#)

Creates a new map route

[**MapRouteMovement**](#) - Class in [movement](#)

Map based movement model that uses predetermined paths within the map area.

[**MapRouteMovement\(Settings\)**](#) - Constructor for class movement.[MapRouteMovement](#)

Creates a new movement model based on a Settings object's settings.

[**MapRouteMovement\(MapRouteMovement\)**](#) - Constructor for class movement.[MapRouteMovement](#)

Copyconstructor.

[**MAX_GROUP_SIZE_SETTING**](#) - Static variable in class movement.[EveningActivityMovement](#)

[**MAX_NODE_COUNT**](#) - Static variable in class gui.[NodeChooser](#)

the maximum number of nodes to show in the list per page

[**MAX_TYPE**](#) - Static variable in class movement.map.[MapNode](#)

Biggest valid type of a node: 31

[**MAX_WAIT_TIME_SETTING**](#) - Static variable in class movement.[EveningActivityMovement](#)

[**maxPathLength**](#) - Variable in class movement.[MapBasedMovement](#)

max nrof map nodes to travel/path

[**MAXPROP_NS**](#) - Static variable in class routing.[MaxPropRouter](#)

Router's setting namespace ("MaxPropRouter")

[**MAXPROP_NS**](#) - Static variable in class routing.[MaxPropRouterWithEstimation](#)

MaxPROP router's setting namespace ("MaxPropRouterWithEstimation")

[**MaxPropDijkstra**](#) - Class in [routing.maxprop](#)

Dijkstra's shortest path implementation for MaxProp Router module.

[**MaxPropDijkstra\(Map<Integer, MeetingProbabilitySet>\)**](#) - Constructor for class

routing.maxprop.[MaxPropDijkstra](#)

Constructor.

[MaxPropRouter](#) - Class in [routing](#)

Implementation of MaxProp router as described in *MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks* by John Burgess et al.

[MaxPropRouter\(Settings\)](#) - Constructor for class routing.[MaxPropRouter](#)
Constructor.**[MaxPropRouter\(MaxPropRouter\)](#)** - Constructor for class routing.[MaxPropRouter](#)
Copy constructor.**[MaxPropRouterWithEstimation](#)** - Class in [routing](#)

Implementation of MaxProp router as described in *MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks* by John Burgess et al.

[MaxPropRouterWithEstimation\(Settings\)](#) - Constructor for class routing.[MaxPropRouterWithEstimation](#)
Constructor.**[MaxPropRouterWithEstimation\(MaxPropRouterWithEstimation\)](#)** - Constructor for class routing.[MaxPropRouterWithEstimation](#)
routing.[MaxPropRouterWithEstimation](#)
Copy constructor.**[maxSpeed](#)** - Variable in class movement.[MovementModel](#)**[maxWaitTime](#)** - Variable in class movement.[MovementModel](#)**[MEETING_SPOTS_FILE_SETTING](#)** - Static variable in class movement.[EveningActivityMovement](#)**[MeetingProbabilitySet](#)** - Class in [routing.maxprop](#)

Class for storing and manipulating the meeting probabilities for the MaxProp router module.

[MeetingProbabilitySet\(int, double\)](#) - Constructor for class routing.maxprop.[MeetingProbabilitySet](#)
Constructor.**[MeetingProbabilitySet\(\)](#)** - Constructor for class routing.maxprop.[MeetingProbabilitySet](#)
Constructor.**[MeetingProbabilitySet\(double, List<Integer>\)](#)** - Constructor for class routing.maxprop.[MeetingProbabilitySet](#)
Constructor.**[Message](#)** - Class in [core](#)

A message that is created at a node or passed between nodes.

[Message\(DTNHost, DTNHost, String, int\)](#) - Constructor for class core.[Message](#)
Creates a new Message.**[MESSAGE_ID_PREFIX_S](#)** - Static variable in class input.[MessageEventGenerator](#)
Message ID prefix -setting id ("prefix").**[MESSAGE_INTERVAL_S](#)** - Static variable in class input.[MessageEventGenerator](#)
Message creation interval range -setting id ("interval").**[MESSAGE_SIZE_S](#)** - Static variable in class input.[MessageEventGenerator](#)
Message size range -setting id ("size").**[MESSAGE_TIME_S](#)** - Static variable in class input.[MessageEventGenerator](#)
Message creation time range -setting id ("time").**[MESSAGE_TRANS_DELIVERED](#)** - Static variable in class report.[EventLogReport](#)
Extra info for message relayed event ("delivered"): "D"**[MESSAGE_TRANS_DELIVERED AGAIN](#)** - Static variable in class report.[EventLogReport](#)
Extra info for message relayed event ("delivered again"): "A"**[MESSAGE_TRANS_RELAYED](#)** - Static variable in class report.[EventLogReport](#)
Extra info for message relayed event ("relayed"): "R"**[messageAborted\(String, DTNHost, int\)](#)** - Method in class core.[DTNHost](#)
Informs the host that a message transfer was aborted.**[messageAborted\(String, DTNHost, int\)](#)** - Method in class routing.[MessageRouter](#)
This method should be called (on the receiving host) when a message transfer was aborted.**[MessageBurstGenerator](#)** - Class in [input](#)

Message creation -external events generator.

[MessageBurstGenerator\(Settings\)](#) - Constructor for class input.[MessageBurstGenerator](#)**[MessageCreateEvent](#)** - Class in [input](#)

External event for creating a message.

[**MessageCreateEvent\(int, int, String, int, int, double\)**](#) - Constructor for class input.[MessageCreateEvent](#)

Creates a message creation event with a optional response request

[**MessageDelayReport**](#) - Class in [report](#)

Reports delivered messages' delays (one line per delivered message) and cumulative delivery probability sorted by message delays.

[**MessageDelayReport\(\)**](#) - Constructor for class report.[MessageDelayReport](#)

Constructor.

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in interface core.[MessageListener](#)

Method is called when a message is deleted

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class gui.[EventLogPanel](#)

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[CreatedMessagesReport](#)

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[DeliveredMessagesReport](#)

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[DistanceDelayReport](#)

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[DTN2Reporter](#)

Method is called when a message is deleted

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[EventLogReport](#)

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[MessageDelayReport](#)

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[MessageDeliveryReport](#)

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[MessageGraphvizReport](#)

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[MessageReport](#)

[**messageDeleted\(Message, DTNHost, boolean\)**](#) - Method in class report.[MessageStatsReport](#)

[**MessageDeleteEvent**](#) - Class in [input](#)

External event for deleting a message.

[**MessageDeleteEvent\(int, String, double, boolean\)**](#) - Constructor for class input.[MessageDeleteEvent](#)

Creates a message delete event

[**MessageDeliveryReport**](#) - Class in [report](#)

Report for of amount of messages delivered vs.

[**MessageDeliveryReport\(\)**](#) - Constructor for class report.[MessageDeliveryReport](#)

Constructor.

[**MessageEvent**](#) - Class in [input](#)

A message related external event

[**MessageEvent\(int, int, String, double\)**](#) - Constructor for class input.[MessageEvent](#)

Creates a message event

[**MessageEventGenerator**](#) - Class in [input](#)

Message creation -external events generator.

[**MessageEventGenerator\(Settings\)**](#) - Constructor for class input.[MessageEventGenerator](#)

Constructor, initializes the interval between events, and the size of messages generated, as well as number of hosts in the network.

[**MessageGraphic**](#) - Class in [gui.playfield](#)

Visualization of a message

[**MessageGraphic\(DTNHost, DTNHost\)**](#) - Constructor for class gui.playfield.[MessageGraphic](#)

[**MessageGraphvizReport**](#) - Class in [report](#)

Creates a graphviz compatible graph of messages that were passed.

[**MessageGraphvizReport\(\)**](#) - Constructor for class report.[MessageGraphvizReport](#)

Constructor.

[**MessageListener**](#) - Interface in [core](#)

Interface for classes that want to be informed about messages between hosts

[**MessageLocationReport**](#) - Class in [report](#)

Message location report.

[**MessageLocationReport\(\)**](#) - Constructor for class report.[MessageLocationReport](#)

Constructor.

[**MessageRelayEvent**](#) - Class in [input](#)

External event for all the stages of relaying a message between two hosts (start and possible abort or delivery).

[**MessageRelayEvent\(int, int, String, double, int\)**](#) - Constructor for class input.[MessageRelayEvent](#)

Creates a message relaying event

[**MessageReport**](#) - Class in [report](#)

Reports delivered messages report: message_id creation_time deliver_time (duplicate)

[**MessageReport\(\)**](#) - Constructor for class report.[MessageReport](#)

Constructor.

[**MessageRouter**](#) - Class in [routing](#)

Superclass for message routers.

[**MessageRouter\(Settings\)**](#) - Constructor for class routing.[MessageRouter](#)

Constructor.

[**MessageRouter\(MessageRouter\)**](#) - Constructor for class routing.[MessageRouter](#)

Copy-constructor.

[**MessageStatsReport**](#) - Class in [report](#)

Report for generating different kind of total statistics about message relaying performance.

[**MessageStatsReport\(\)**](#) - Constructor for class report.[MessageStatsReport](#)

Constructor.

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in interface core.[MessageListener](#)

Method is called when a message's transfer was aborted before it finished

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class gui.[EventLogPanel](#)

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[CreatedMessagesReport](#)

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[DeliveredMessagesReport](#)

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[DistanceDelayReport](#)

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[DTN2Reporter](#)

Method is called when a message's transfer was aborted before it finished

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[EventLogReport](#)

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[MessageDelayReport](#)

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[MessageDeliveryReport](#)

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[MessageGraphvizReport](#)

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[MessageReport](#)

[**messageTransferAborted\(Message, DTNHost, DTNHost\)**](#) - Method in class report.[MessageStatsReport](#)

[**messageTransferred\(String, DTNHost\)**](#) - Method in class core.[DTNHost](#)

Informs the host that a message was successfully transferred.

[**messageTransferred\(Message, DTNHost, DTNHost, boolean\)**](#) - Method in interface core.[MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

[**messageTransferred\(Message, DTNHost, DTNHost, boolean\)**](#) - Method in class gui.[EventLogPanel](#)

[**messageTransferred\(Message, DTNHost, DTNHost, boolean\)**](#) - Method in class report.[CreatedMessagesReport](#)

[**messageTransferred\(Message, DTNHost, DTNHost, boolean\)**](#) - Method in class

report.[DeliveredMessagesReport](#)

[messageTransferred\(Message, DTNHost, DTNHost, boolean\)](#) - Method in class report.[DistanceDelayReport](#)

This is called when a message is transferred between nodes

[messageTransferred\(Message, DTNHost, DTNHost, boolean\)](#) - Method in class report.[DTN2Reporter](#)

Method is called when a message is successfully transferred from a node to another.

[messageTransferred\(Message, DTNHost, DTNHost, boolean\)](#) - Method in class report.[EventLogReport](#)

[messageTransferred\(Message, DTNHost, DTNHost, boolean\)](#) - Method in class report.[MessageDelayReport](#)

[messageTransferred\(Message, DTNHost, DTNHost, boolean\)](#) - Method in class report.[MessageDeliveryReport](#)

[messageTransferred\(Message, DTNHost, DTNHost, boolean\)](#) - Method in class report.[MessageGraphvizReport](#)

[messageTransferred\(Message, DTNHost, DTNHost, boolean\)](#) - Method in class report.[MessageReport](#)

[messageTransferred\(Message, DTNHost, DTNHost, boolean\)](#) - Method in class report.[MessageStatsReport](#)

[messageTransferred\(String, DTNHost\)](#) - Method in class routing.[ActiveRouter](#)

[messageTransferred\(String, DTNHost\)](#) - Method in class routing.[EpidemicOracleRouter](#)

[messageTransferred\(String, DTNHost\)](#) - Method in class routing.[MaxPropRouter](#)

[messageTransferred\(String, DTNHost\)](#) - Method in class routing.[MaxPropRouterWithEstimation](#)

[messageTransferred\(String, DTNHost\)](#) - Method in class routing.[MessageRouter](#)

This method should be called (on the receiving host) after a message was successfully transferred.

[messageTransferred\(String, DTNHost\)](#) - Method in class routing.[SprayAndWaitRouter](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in interface core.[MessageListener](#)

Method is called when a message's transfer is started

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class gui.[EventLogPanel](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[CreatedMessagesReport](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[DeliveredMessagesReport](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[DistanceDelayReport](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[DTN2Reporter](#)

Method is called when a message's transfer is started

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[EventLogReport](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[MessageDelayReport](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[MessageDeliveryReport](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[MessageGraphvizReport](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[MessageReport](#)

[messageTransferStarted\(Message, DTNHost, DTNHost\)](#) - Method in class report.[MessageStatsReport](#)

[MIN_GROUP_SIZE_SETTING](#) - Static variable in class movement.[EveningActivityMovement](#)

[MIN_TYPE](#) - Static variable in class movement.map.[MapNode](#)

Smallest valid type of a node: 1

[MIN_WAIT_TIME_SETTING](#) - Static variable in class movement.[EveningActivityMovement](#)

[minPathLength](#) - Variable in class movement.[MapBasedMovement](#)

min nrof map nodes to travel/path

[minSpeed](#) - Variable in class movement.[MovementModel](#)

[minWaitTime](#) - Variable in class movement.[MovementModel](#)

[mirror\(\)](#) - Method in class movement.map.[SimMap](#)

Mirrors all map coordinates around X axis ($x'=x$, $y'=-y$).

[MM_WARMUP_S](#) - Static variable in class ui.[DTNSimUI](#)

Movement model warmup time -setting id ("MovementModel.warmup").

[ModuleCommunicationBus](#) - Class in [core](#)

Intermodule communication bus.

[ModuleCommunicationBus\(\)](#) - Constructor for class core.[ModuleCommunicationBus](#)

Constructor.

[ModuleCommunicationListener](#) - Interface in [core](#)

This interface should be implemented by classes that want to be notified of variable value changes in ModuleCommunicationBuses.

[moduleValueChanged\(String, Object\)](#) - Method in interface core.[ModuleCommunicationListener](#)

This method is called whenever a variable, whose changes the module has registered to, changes.

[moduleValueChanged\(String, Object\)](#) - Method in class core.[NetworkInterface](#)

This method is called by the [ModuleCommunicationBus](#) when/if someone changes the scanning interval, transmit speed, or range

[moduleValueChanged\(String, Object\)](#) - Method in class routing.[EnergyAwareRouter](#)

Called by the combus is the energy value is changed

[move\(double\)](#) - Method in class core.[DTNHost](#)

Moves the node towards the next waypoint or waits if it is not time to move yet

[moveInterface\(NetworkInterface, ConnectivityGrid.GridCell\)](#) - Method in class interfaces.[ConnectivityGrid.GridCell](#)

Moves a interface in a Cell to another Cell

[movement](#) - package movement

Contains different movement models and related classes for the simulator.

[movement.map](#) - package movement.map

Sub package for MapBasedMovement movement model's (and its sub classes) helper classes.

[MOVEMENT_FILE_S](#) - Static variable in class movement.[ExternalMovement](#)

external locations file's path -setting id ("file")

[MOVEMENT_MODEL_NS](#) - Static variable in class movement.[MovementModel](#)

MovementModel namespace (where world size and rng seed settings are looked from ("MovementModel"))

[MOVEMENT_MODEL_S](#) - Static variable in class core.[SimScenario](#)

movement model class -setting id ("movementModel")

[MovementListener](#) - Interface in [core](#)

Interface for classes that want to be informed about node movement.

[MovementModel](#) - Class in [movement](#)

Superclass for all movement models.

[MovementModel\(\)](#) - Constructor for class movement.[MovementModel](#)

Empty constructor for testing purposes.

[MovementModel\(Settings\)](#) - Constructor for class movement.[MovementModel](#)

Creates a new MovementModel based on a Settings object's settings.

[MovementModel\(MovementModel\)](#) - Constructor for class movement.[MovementModel](#)

Copyconstructor.

[MovementNs2Report](#) - Class in [report](#)

Movement report that generates suitable movement data for ns-2 simulator as described in

<http://www.isi.edu/nsnam/ns/doc/node174.html>.

[MovementNs2Report\(\)](#) - Constructor for class report.[MovementNs2Report](#)

Constructor.

[**MSG_COUNT_PROPERTY**](#) - Static variable in class routing.[SprayAndWaitRouter](#)

Message property key

[**MSG_TTL_S**](#) - Static variable in class routing.[MessageRouter](#)

Message TTL -setting id ("msgTtl").

[**msgFromNode**](#) - Variable in class core.[Connection](#)

[**msgOnFly**](#) - Variable in class core.[Connection](#)

[**msgTime**](#) - Variable in class input.[MessageEventGenerator](#)

Time range for message creation (min, max)

[**msgTtl**](#) - Variable in class routing.[MessageRouter](#)

TTL for all messages

[**MULTILINESTRING**](#) - Static variable in class input.[WKTReader](#)

known WKT type MULTILINESTRING

N

[**NAME_S**](#) - Static variable in class core.[SimScenario](#)

scenario name -setting id ("name")

[**NAN**](#) - Static variable in class report.[Report](#)

String value for values that could not be calculated

[**NetworkInterface**](#) - Class in [core](#)

Network interface of a DTNHost.

[**NetworkInterface\(Settings\)**](#) - Constructor for class core.[NetworkInterface](#)

For creating an empty class of a specific type

[**NetworkInterface\(\)**](#) - Constructor for class core.[NetworkInterface](#)

For creating an empty class of a specific type

[**NetworkInterface\(NetworkInterface\)**](#) - Constructor for class core.[NetworkInterface](#)

copy constructor

[**newDestination\(DTNHost, Coord, double\)**](#) - Method in interface core.[MovementListener](#)

Method is called every time a host receives a new destination from its movement model.

[**newDestination\(DTNHost, Coord, double\)**](#) - Method in class report.[MovementNs2Report](#)

[**newEvent\(\)**](#) - Method in class report.[Report](#)

This method should be called before every new (complete) event the report logs.

[**newMessage\(Message\)**](#) - Method in interface core.[MessageListener](#)

Method is called when a new message is created

[**newMessage\(Message\)**](#) - Method in class gui.[EventLogPanel](#)

[**newMessage\(Message\)**](#) - Method in class report.[CreatedMessagesReport](#)

[**newMessage\(Message\)**](#) - Method in class report.[DeliveredMessagesReport](#)

[**newMessage\(Message\)**](#) - Method in class report.[DistanceDelayReport](#)

This is called when a new message is created

[**newMessage\(Message\)**](#) - Method in class report.[DTN2Reporter](#)

Method is called when a new message is created

[**newMessage\(Message\)**](#) - Method in class report.[EventLogReport](#)

[**newMessage\(Message\)**](#) - Method in class report.[MessageDelayReport](#)

[**newMessage\(Message\)**](#) - Method in class report.[MessageDeliveryReport](#)

[**newMessage\(Message\)**](#) - Method in class report.[MessageGraphvizReport](#)

[**newMessage\(Message\)**](#) - Method in class report.[MessageReport](#)

[**newMessage\(Message\)**](#) - Method in class report.[MessageStatsReport](#)

[**newOrders\(\)**](#) - Method in class movement.[ExtendedMovementModel](#)

Method is called between each getPath() request when the current MM is ready (isReady() method returns true).

[**newOrders\(\)**](#) - Method in class movement.[WorkingDayMovement](#)

[**nextEvent\(\)**](#) - Method in class input.[DTN2Events](#)

[**nextEvent\(\)**](#) - Method in interface input.[EventQueue](#)

Returns the next event in the queue or ExternalEvent with time of double.MAX_VALUE if there are no events left.

[**nextEvent\(\)**](#) - Method in class input.[ExternalEventsQueue](#)

Returns the next event in the queue or ExternalEvent with time of double.MAX_VALUE if there are no events left

[**nextEvent\(\)**](#) - Method in class input.[MessageBurstGenerator](#)

Returns the next message creation event

[**nextEvent\(\)**](#) - Method in class input.[MessageEventGenerator](#)

Returns the next message creation event

[**nextEvent\(\)**](#) - Method in class input.[OneFromEachMessageGenerator](#)

Returns the next message creation event

[**nextEvent\(\)**](#) - Method in class input.[OneToEachMessageGenerator](#)

Returns the next message creation event

[**nextEvent\(\)**](#) - Method in class input.[ScheduledUpdatesQueue](#)

Returns the next scheduled event or event with time Double.MAX_VALUE if there aren't any.

[**nextEventsTime\(\)**](#) - Method in class input.[DTN2Events](#)

[**nextEventsTime\(\)**](#) - Method in interface input.[EventQueue](#)

Returns next event's time or Double.MAX_VALUE if there are no events left in the queue.

[**nextEventsTime\(\)**](#) - Method in class input.[ExternalEventsQueue](#)

Returns next event's time or Double.MAX_VALUE if there are no events left

[**nextEventsTime**](#) - Variable in class input.[MessageEventGenerator](#)

Time of the next event (simulated seconds)

[**nextEventsTime\(\)**](#) - Method in class input.[MessageEventGenerator](#)

Returns next message creation event's time

[**nextEventsTime\(\)**](#) - Method in class input.[ScheduledUpdatesQueue](#)

Returns the next scheduled event's time or Double.MAX_VALUE if there aren't any events left

[**nextPathAvailable\(\)**](#) - Method in class movement.[ExternalMovement](#)

Returns a sim time when the next path is available.

[**nextPathAvailable\(\)**](#) - Method in class movement.[LinearFormation](#)

Returns Double.MAX_VALUE (no paths available)

[**nextPathAvailable\(\)**](#) - Method in class movement.[MovementModel](#)

Returns a sim time when the next path is available.

[**nextPathAvailable\(\)**](#) - Method in class movement.[StationaryMovement](#)

[**nextStop\(\)**](#) - Method in class movement.map.[MapRoute](#)

Returns the next stop on the route (depending on the route mode)

[**nextType\(\)**](#) - Method in class input.[WKTReader](#)

Returns the next type read from the reader given at init or null if no more types can be read

[**NODE_ARR_S**](#) - Static variable in class report.[MovementNs2Report](#)

node array's name -setting id ("nodeArray")

[**NodeChooser**](#) - Class in [gui](#)

Node chooser panel

[**NodeChooser\(List<DTNHost>, DTNSimGUI\)**](#) - Constructor for class [gui.NodeChooser](#)

[**NodeGraphic**](#) - Class in [gui.playfield](#)

Visualization of a DTN Node

[**NodeGraphic\(DTNHost\)**](#) - Constructor for class gui.playfield.[NodeGraphic](#)

[**NR_OF_MEETING_SPOTS_SETTING**](#) - Static variable in class movement.[EveningActivityMovement](#)

[**NR_OF_OFFICES_SETTING**](#) - Static variable in class movement.[OfficeActivityMovement](#)

[**NROF_COPIES**](#) - Static variable in class routing.[SprayAndWaitRouter](#)

identifier for the initial number of copies setting ("nrofCopies")

[**NROF_FILES_S**](#) - Static variable in class movement.[MapBasedMovement](#)

number of map files -setting id ("nrofMapFiles")

[**NROF_GROUPS_S**](#) - Static variable in class core.[SimScenario](#)

number of host groups -setting id ("nrofHostGroups")

[**NROF_HOSTS_S**](#) - Static variable in class core.[SimScenario](#)

number of hosts in the group -setting id ("nrofHosts")

[**NROF_INTERF_S**](#) - Static variable in class core.[SimScenario](#)

number of interfaces in the group -setting id ("nrofInterfaces")

[**NROF_INTTYPES_S**](#) - Static variable in class core.[SimScenario](#)

number of interface types -setting id ("nrofInterfaceTypes")

[**NROF_PRELOAD_S**](#) - Static variable in class movement.[ExternalMovement](#)

number of preloaded intervals per preload run -setting id ("nrofPreload")

[**NROF_REPORT_S**](#) - Static variable in class ui.[DTNSimUI](#)

Number of reports -setting id ("Report.nrofReports").

[**NROF_SETTING**](#) - Static variable in class input.[EventQueueHandler](#)

number of event queues -setting id ("nrof")

[**NS_CMD_S**](#) - Static variable in class report.[MovementNs2Report](#)

ns command -setting id ("nsCmd")

[**numberOfTransmissions**](#) - Variable in class interfaces.[InterferenceLimitedInterface](#)

O

[**OFFICE_LOCATIONS_FILE_SETTING**](#) - Static variable in class movement.[OfficeActivityMovement](#)

[**OFFICE_MAX_WAIT_TIME_SETTING**](#) - Static variable in class movement.[OfficeActivityMovement](#)

[**OFFICE_MIN_WAIT_TIME_SETTING**](#) - Static variable in class movement.[OfficeActivityMovement](#)

[**OFFICE_SIZE_SETTING**](#) - Static variable in class movement.[OfficeActivityMovement](#)

[**OFFICE_WAIT_TIME_PARETO_COEFF_SETTING**](#) - Static variable in class movement.[OfficeActivityMovement](#)

[**OfficeActivityMovement**](#) - Class in [movement](#)

This class models movement at an office.

[**OfficeActivityMovement\(Settings\)**](#) - Constructor for class movement.[OfficeActivityMovement](#)

OfficeActivityMovement constructor

[**OfficeActivityMovement\(OfficeActivityMovement\)**](#) - Constructor for class movement.[OfficeActivityMovement](#)

Copyconstructor

[**OneFromEachMessageGenerator**](#) - Class in [input](#)

Message creation -external events generator.

[**OneFromEachMessageGenerator\(Settings\)**](#) - Constructor for class input.[OneFromEachMessageGenerator](#)

[**OneToEachMessageGenerator**](#) - Class in [input](#)

Message creation -external events generator.

[**OneToEachMessageGenerator\(Settings\)**](#) - Constructor for class input.[OneToEachMessageGenerator](#)

[**optimizer**](#) - Variable in class core.[NetworkInterface](#)

[**out**](#) - Variable in class report.[Report](#)

The print writer used to write output.

[**OUT_SUFFIX**](#) - Static variable in class report.[Report](#)

Suffix of report files without explicit output

[**OUTPUT_SETTING**](#) - Static variable in class report.[Report](#)

The output file path of the report -setting id ("output")

P

[**p\(String\)**](#) - Static method in class core.[Debug](#)

Prints text to output with level 0

[**p\(String, int\)**](#) - Static method in class core.[Debug](#)

Prints text to output given with level

[**p\(String, int, boolean\)**](#) - Static method in class core.[Debug](#)

Print text to debug output.

[**P_AVG_TARGET_S**](#) - Static variable in class routing.[ProphetRouterWithEstimation](#)

Target P_avg

[**P_INIT**](#) - Static variable in class routing.[ProphetRouter](#)

delivery predictability initialization constant

[**P_INIT**](#) - Static variable in class routing.[ProphetRouterWithEstimation](#)

delivery predictability initialization constant

[**paint\(Graphics\)**](#) - Method in class gui.playfield.[PlayField](#)

Draws the play field.

[**ParetoRNG**](#) - Class in [core](#)

A random number generator for a Pareto distribution

[**ParetoRNG\(Random, double, double, double\)**](#) - Constructor for class core.[ParetoRNG](#)

Creates a new Pareto random number generator that makes use of a normal random number generator

[**parseError\(String\)**](#) - Method in class input.[DTN2Events.ParserHandler](#)

[**parseLineString\(String\)**](#) - Method in class input.[WKTReader](#)

Parses coordinate tuples from "LINESTRING" lines

[**parseMultilinestring\(\)**](#) - Method in class input.[WKTReader](#)

Parses a MULTILINESTRING statement that has nested linestrings from the current reader

[**parsePoint\(\)**](#) - Method in class input.[WKTReader](#)

Parses a WKT point data from the initialized reader

[**PassiveRouter**](#) - Class in [routing](#)

Passive router that doesn't send anything unless commanded.

[**PassiveRouter\(Settings\)**](#) - Constructor for class routing.[PassiveRouter](#)

[**PassiveRouter\(PassiveRouter\)**](#) - Constructor for class routing.[PassiveRouter](#)

Copy-constructor.

[**Path**](#) - Class in [movement](#)

A Path between multiple Coordinates.

[**Path\(\)**](#) - Constructor for class movement.[Path](#)

Creates a path with zero speed.

[**Path\(Path\)**](#) - Constructor for class movement.[Path](#)

Copy constructor.

[**Path\(double\)**](#) - Constructor for class movement.[Path](#)

Creates a path with constant speed

[**PATH_SETTING**](#) - Static variable in class input.[EventQueueHandler](#)

path of external events file -setting id ("filePath")

[**PATH_SETTING**](#) - Static variable in class input.[ExternalEventsQueue](#)

path of external events file -setting id ("filePath")

[**PathGraphic**](#) - Class in [gui.playfield](#)

Visualization of a Path

[**PathGraphic\(Path\)**](#) - Constructor for class [gui.playfield.PathGraphic](#)

[**pauseOnEvent\(\)**](#) - Method in class [gui.EventLogControl](#)

Returns true if this event type should cause pause

[**PING_DEST_RANGE**](#) - Static variable in class [applications.PingApplication](#)

Destination address range - inclusive lower, exclusive upper

[**PING_INTERVAL**](#) - Static variable in class [applications.PingApplication](#)

Ping generation interval

[**PING_OFFSET**](#) - Static variable in class [applications.PingApplication](#)

Ping interval offset - avoids synchronization of ping sending

[**PING_PASSIVE**](#) - Static variable in class [applications.PingApplication](#)

Run in passive mode - don't generate pings but respond

[**PING_PING_SIZE**](#) - Static variable in class [applications.PingApplication](#)

Size of the ping message

[**PING_PONG_SIZE**](#) - Static variable in class [applications.PingApplication](#)

Size of the pong message

[**PING_SEED**](#) - Static variable in class [applications.PingApplication](#)

Seed for the app's random number generator

[**PingApplication**](#) - Class in [applications](#)

Simple ping application to demonstrate the application support.

[**PingApplication\(Settings\)**](#) - Constructor for class [applications.PingApplication](#)

Creates a new ping application with the given settings.

[**PingApplication\(PingApplication\)**](#) - Constructor for class [applications.PingApplication](#)

Copy-constructor

[**PingAppReporter**](#) - Class in [report](#)

Reporter for the [PingApplication](#).

[**PingAppReporter\(\)**](#) - Constructor for class [report.PingAppReporter](#)

[**PINGPONG**](#) - Static variable in class [movement.map.MapRoute](#)

Type of the route ID: ping-pong (2).

[**PlayField**](#) - Class in [gui.playfield](#)

The canvas where node graphics and message visualizations are drawn.

[**PlayField\(World\)**](#) - Constructor for class [gui.playfield.PlayField](#)

Creates a playfield

[**PlayFieldGraphic**](#) - Class in [gui.playfield](#)

Superclass for all graphics to be drawn on the "play field".

[**PlayFieldGraphic\(\)**](#) - Constructor for class [gui.playfield.PlayFieldGraphic](#)

[**POI_FILE_S**](#) - Static variable in class [movement.map.PointsOfInterest](#)

Points Of Interest file path -prefix id ("poiFile")

[**POI_NS**](#) - Static variable in class [movement.map.PointsOfInterest](#)

Points Of Interest settings namespace ("PointsOfInterest")

[**POI_SELECT_S**](#) - Static variable in class [movement.map.PointsOfInterest](#)

Per node group setting used for selecting POI groups and their probabilities ("pois").

Syntax: poiGroupIndex1, groupSelectionProbability1, groupIndex2, prob2, etc...

Sum of probabilities must be less than or equal to one (1.0).

[**POINT**](#) - Static variable in class [input.WKTReader](#)

known WKT type POINT

[**PointsOfInterest**](#) - Class in [movement.map](#)

Handler for points of interest data.

[**PointsOfInterest\(SimMap, int\[\], Settings, Random\)**](#) - Constructor for class [movement.map.PointsOfInterest](#)

Constructor.

[**PRECISION_SETTING**](#) - Static variable in class [report.Report](#)

Precision of formatted double values - setting id ("precision").

[**PRELOAD_SETTING**](#) - Static variable in class [input.EventQueueHandler](#)

number of events to preload from file -setting id ("nrofPreload")

PRELOAD_SETTING - Static variable in class input.[ExternalEventsQueue](#)

number of event to preload -setting id ("nrofPreload")

PROB_SET_MAX_SIZE_S - Static variable in class routing.[MaxPropRouter](#)

Meeting probability set maximum size -setting id ("probSetMaxSize").

PROBABILITIES_STRING - Static variable in class movement.[BusTravellerMovement](#)

PROBABILITY TAKE OTHER BUS - Static variable in class movement.[BusTravellerMovement](#)

PROBABILITY TO GO SHOPPING SETTING - Static variable in class movement.[WorkingDayMovement](#)

PROBABILITY TO OWN CAR SETTING - Static variable in class movement.[WorkingDayMovement](#)

processEvent(World) - Method in class input.[ConnectionEvent](#)

processEvent(World) - Method in class input.[ExternalEvent](#)

Processes the external event.

processEvent(World) - Method in class input.[MessageCreateEvent](#)

Creates the message this event represents.

processEvent(World) - Method in class input.[MessageDeleteEvent](#)

Deletes the message

processEvent(World) - Method in class input.[MessageRelayEvent](#)

Relays the message

PROPHET_NS - Static variable in class routing.[ProphetRouter](#)

Prophet router's setting namespace ("ProphetRouter")

PROPHET_NS - Static variable in class routing.[ProphetRouterWithEstimation](#)

Prophet router's setting namespace ("ProphetRouterWithEstimation")

ProphetRouter - Class in [routing](#)

Implementation of PROPHET router as described in *Probabilistic routing in intermittently connected networks* by Anders Lindgren et al.

ProphetRouter(Settings) - Constructor for class routing.[ProphetRouter](#)

Constructor.

ProphetRouter(ProphetRouter) - Constructor for class routing.[ProphetRouter](#)

Copyconstructor.

ProphetRouterWithEstimation - Class in [routing](#)

Implementation of PROPHET router as described in *Probabilistic routing in intermittently connected networks* by Anders Lindgren et al.

ProphetRouterWithEstimation(Settings) - Constructor for class routing.[ProphetRouterWithEstimation](#)

Constructor.

ProphetRouterWithEstimation(ProphetRouterWithEstimation) - Constructor for class

routing.[ProphetRouterWithEstimation](#)

Copyconstructor.

props - Static variable in class core.[Settings](#)

properties object where the setting files are read into

pt(String, int) - Static method in class core.[Debug](#)

Debug print with a timestamp

pt(String) - Static method in class core.[Debug](#)

Debug print with a timestamp and 0 level

putToIncomingBuffer(Message, DTNHost) - Method in class routing.[MessageRouter](#)

Puts a message to incoming messages buffer.

Q

Q_MODE_FIFO - Static variable in class routing.[MessageRouter](#)

Setting value for FIFO queue mode

Q_MODE_RANDOM - Static variable in class routing.[MessageRouter](#)

Setting value for random queue mode

R

[**randomCoord\(\)**](#) - Method in class movement.[ClusterMovement](#)

[**randomCoord\(\)**](#) - Method in class movement.[RandomWaypoint](#)

[**RANDOMIZE_UPDATES_S**](#) - Static variable in class core.[World](#)

Should the order of node updates be different (random) within every update step -setting id ("randomizeUpdateOrder").

[**RandomWalk**](#) - Class in [movement](#)

Random Walk movement model

[**RandomWalk\(Settings\)**](#) - Constructor for class movement.[RandomWalk](#)

[**RandomWaypoint**](#) - Class in [movement](#)

Random waypoint movement model.

[**RandomWaypoint\(Settings\)**](#) - Constructor for class movement.[RandomWaypoint](#)

[**RandomWaypoint\(RandomWaypoint\)**](#) - Constructor for class movement.[RandomWaypoint](#)

[**RANGE_DELIMETER**](#) - Static variable in class core.[DTNSim](#)

Delimiter for batch mode index range values (colon)

[**RANGE_ID**](#) - Static variable in class core.[NetworkInterface](#)

[ModuleCommunicationBus](#) identifier for the "radio range" variable.

[**RCV_OK**](#) - Static variable in class routing.[MessageRouter](#)

Receive return value for OK

[**readEvents\(int\)**](#) - Method in class input.[BinaryEventsReader](#)

Read events from a binary file created with storeBinaryFile method

[**readEvents\(int\)**](#) - Method in interface input.[ExternalEventsReader](#)

Read events from the reader

[**readEvents\(int\)**](#) - Method in class input.[StandardEventsReader](#)

[**readLines\(File\)**](#) - Method in class input.[WKTRader](#)

Read line (LINESTRING) data from a file

[**readNestedContents\(Reader\)**](#) - Method in class input.[WKTRader](#)

Reads everything from the first opening parenthesis until line that ends to a closing parenthesis and returns the contents in one string

[**readNestedContents\(\)**](#) - Method in class input.[WKTRader](#)

Returns nested contents from the reader given at init

[**readNextMovements\(\)**](#) - Method in class input.[ExternalMovementReader](#)

Reads all new id-coordinate tuples that belong to the same time instance

[**readPoints\(File\)**](#) - Method in class input.[WKTRader](#)

Read point data from a file

[**readPoints\(Reader\)**](#) - Method in class input.[WKTRader](#)

Read point data from a Reader

[**readRoutes\(String, int, SimMap\)**](#) - Static method in class movement.map.[MapRoute](#)

Reads routes from files defined in Settings

[**readWord\(Reader\)**](#) - Method in class input.[WKTRader](#)

Reads a "word", ie whitespace delimited string of characters, from the reader

[**receiveMessage\(Message, DTNHost\)**](#) - Method in class core.[DTNHost](#)

Start receiving a message from another host

[**receiveMessage\(Message, DTNHost\)**](#) - Method in class routing.[ActiveRouter](#)

[**receiveMessage\(Message, DTNHost\)**](#) - Method in class routing.[MessageRouter](#)

Try to start receiving a message from another host.

[**receiveMessage\(Message, DTNHost\)**](#) - Method in class routing.[SprayAndWaitRouter](#)

[**reduceEnergy\(double\)**](#) - Method in class routing.[EnergyAwareRouter](#)

Updates the current energy so that the given amount is reduced from it.

[**reduceSendingAndScanningEnergy\(\)**](#) - Method in class routing.[EnergyAwareRouter](#)

Reduces the energy reserve for the amount that is used by sending data and scanning for the other nodes.

[**registerBus\(BusMovement\)**](#) - Method in class movement.[BusControlSystem](#)

Registers a bus to be part of a bus control system

[**registerForReset\(String\)**](#) - Static method in class core.[DTNSim](#)

Registers a class for resetting.

[**registerTraveller\(BusTravellerMovement\)**](#) - Method in class movement.[BusControlSystem](#)

Registers a traveller/passenger to be part of a bus control system

[**REMOVE**](#) - Static variable in class input.[StandardEventsReader](#)

Identifier of message removed event ("R")

[**removeConnection\(DTNHost, DTNHost\)**](#) - Method in class report.[ContactTimesReport](#)

[**removeDeliveredMessage\(String\)**](#) - Method in class routing.[EpidemicOracleRouter](#)

Removes the message with the given ID from this router, if the router has that message; otherwise does nothing.

[**removeFromIncomingBuffer\(String, DTNHost\)**](#) - Method in class routing.[MessageRouter](#)

Removes and returns a message with a certain ID from the incoming messages buffer or null if such message wasn't found.

[**removeFromMessages\(String\)**](#) - Method in class routing.[MessageRouter](#)

Removes and returns a message from the message buffer.

[**removeInterface\(NetworkInterface\)**](#) - Method in class interfaces.[ConnectivityGrid.GridCell](#)

Removes an interface from this cell

[**removeInterface\(NetworkInterface\)**](#) - Method in class interfaces.[ConnectivityGrid](#)

Removes a network interface from the overlay grid

[**removeWarmupID\(String\)**](#) - Method in class report.[Report](#)

Removes a warm up ID from the warm up ID set

[**replicate\(\)**](#) - Method in class applications.[PingApplication](#)

[**replicate\(\)**](#) - Method in class core.[Application](#)

[**replicate\(\)**](#) - Method in class core.[Message](#)

Returns a replicate of this message (identical except for the unique id)

[**replicate\(\)**](#) - Method in class core.[NetworkInterface](#)

Replication function

[**replicate\(\)**](#) - Method in class interfaces.[InterferenceLimitedInterface](#)

[**replicate\(\)**](#) - Method in class interfaces.[SimpleBroadcastInterface](#)

[**replicate\(\)**](#) - Method in class movement.[BusMovement](#)

[**replicate\(\)**](#) - Method in class movement.[BusTravellerMovement](#)

[**replicate\(\)**](#) - Method in class movement.[ClusterMovement](#)

[**replicate\(\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**replicate\(\)**](#) - Method in class movement.[ExternalMovement](#)

[**replicate\(\)**](#) - Method in class movement.[HomeActivityMovement](#)

[**replicate\(\)**](#) - Method in class movement.[LinearFormation](#)

[**replicate\(\)**](#) - Method in class movement.map.[MapRoute](#)

Returns a new route with the same settings

[replicate\(\)](#) - Method in class movement.[MapBasedMovement](#)

[replicate\(\)](#) - Method in class movement.[MapRouteMovement](#)

[replicate\(\)](#) - Method in class movement.[MovementModel](#)

Creates a replicate of the movement model.

[replicate\(\)](#) - Method in class movement.[OfficeActivityMovement](#)

[replicate\(\)](#) - Method in class movement.[RandomWalk](#)

[replicate\(\)](#) - Method in class movement.[RandomWaypoint](#)

[replicate\(\)](#) - Method in class movement.[ShortestPathMapBasedMovement](#)

[replicate\(\)](#) - Method in class movement.[StationaryMovement](#)

[replicate\(\)](#) - Method in class movement.[WorkingDayMovement](#)

[replicate\(\)](#) - Method in class routing.[DirectDeliveryRouter](#)

[replicate\(\)](#) - Method in class routing.[EnergyAwareRouter](#)

[replicate\(\)](#) - Method in class routing.[EpidemicOracleRouter](#)

[replicate\(\)](#) - Method in class routing.[EpidemicRouter](#)

[replicate\(\)](#) - Method in class routing.[FirstContactRouter](#)

[replicate\(\)](#) - Method in class routing.maxprop.[MeetingProbabilitySet](#)

Returns a deep copy of the probability set

[replicate\(\)](#) - Method in class routing.[MaxPropRouter](#)

[replicate\(\)](#) - Method in class routing.[MaxPropRouterWithEstimation](#)

[replicate\(\)](#) - Method in class routing.[MessageRouter](#)

Creates a replicate of this router.

[replicate\(\)](#) - Method in class routing.[PassiveRouter](#)

[replicate\(\)](#) - Method in class routing.[ProphetRouter](#)

[replicate\(\)](#) - Method in class routing.[ProphetRouterWithEstimation](#)

[replicate\(\)](#) - Method in class routing.[SprayAndWaitRouter](#)

[report](#) - package report

Contains all the report classes.

[Report](#) - Class in [report](#)

Abstract superclass for all reports.

[Report\(\)](#) - Constructor for class report.[Report](#)

Constructor.

[REPORT_NS](#) - Static variable in class report.[Report](#)

Name space of the settings that are common to all reports ("Report").

[REPORT_S](#) - Static variable in class ui.[DTNSimUI](#)

Report class name -setting id prefix ("Report.report").

[REPORTDIR_SETTING](#) - Static variable in class report.[Report](#)

The default output directory of reports (can be overridden per report with [Report.OUTPUT_SETTING](#)) -setting

`id ("Report.reportDir")`

REPORTED MESSAGES - Static variable in class report.[MessageLocationReport](#)

Reported messages -setting id ("messages").

REPORTED NODES - Static variable in class report.[EnergyLevelReport](#)

Optional reported nodes (comma separated list of network addresses).

reportedMessages - Variable in class report.[MessageLocationReport](#)

Identifiers of the message which are reported

reportedNodes - Variable in class report.[EnergyLevelReport](#)

Networks addresses (integers) of the nodes which are reported

reports - Variable in class ui.[DTNSimUI](#)

Reports that are loaded for this simulation

requestDeliverableMessages(Connection) - Method in class core.[DTNHost](#)

Requests for deliverable message from this host to be sent trough a connection.

requestDeliverableMessages(Connection) - Method in class routing.[ActiveRouter](#)

requestDeliverableMessages(Connection) - Method in class routing.[MessageRouter](#)

Requests for deliverable message from this router to be sent trough a connection.

reset() - Static method in class core.[DTNHost](#)

Reset the host and its interfaces

reset() - Static method in class core.[Message](#)

Resets all static fields to default values

reset() - Static method in class core.[NetworkInterface](#)

Resets the static fields of the class

reset() - Static method in class core.[SimClock](#)

Resets the static fields of the class

reset() - Static method in class core.[SimScenario](#)

reset() - Static method in class interfaces.[ConnectivityGrid](#)

reset() - Static method in class movement.[BusControlSystem](#)

reset() - Static method in class movement.[BusTravellerMovement](#)

reset() - Static method in class movement.[EveningActivityControlSystem](#)

reset() - Static method in class movement.[EveningActivityMovement](#)

reset() - Static method in class movement.[ExternalMovement](#)

Reset state so that next instance will have a fresh state

reset() - Static method in class movement.[MovementModel](#)

Resets all static fields to default values

reset() - Static method in class routing.[EpidemicOracleRouter](#)

Resets the static router list

RESET_METHOD_NAME - Static variable in class core.[DTNSim](#)

Name of the static method that all resettable classes must have

RESPONSE_PREFIX - Static variable in class routing.[ActiveRouter](#)

prefix of all response message IDs

restoreNameSpace() - Method in class core.[Settings](#)

Restores the namespace that was in use before a call to setNameSpace

restoreSecondaryNamespace() - Method in class core.[Settings](#)

Restores the secondary namespace that was in use before a call to setSecondaryNameSpace

rng - Variable in class input.[MessageEventGenerator](#)

Random number generator for this Class

rng - Static variable in class movement.[MovementModel](#)

common rng for all movement models in the simulation

RNG_SEED - Static variable in class movement.[MovementModel](#)

movement models' rng seed -setting id ("rngSeed")

[ROUTE_FILE_S](#) - Static variable in class movement.[MapRouteMovement](#)

Per node group setting used for selecting a route file ("routeFile")

[ROUTE_FIRST_STOP_S](#) - Static variable in class movement.[MapRouteMovement](#)

Per node group setting for selecting which stop (counting from 0 from the start of the route) should be the first one.

[ROUTE_TYPE_S](#) - Static variable in class movement.[MapRouteMovement](#)

Per node group setting used for selecting a route's type ("routeType").

[ROUTER_S](#) - Static variable in class core.[SimScenario](#)

router class -setting id ("router")

[routing](#) - package routing

Contains all the router classes who decide how to handle the messages.

[routing.maxprop](#) - package routing.maxprop

Contains MaxProp routing module specific classes.

[routing.schedule](#) - package routing.schedule

[RoutingInfo](#) - Class in [routing](#)

Class for storing routing related information in a tree form for user interface(s).

[RoutingInfo\(String\)](#) - Constructor for class [routing.RoutingInfo](#)

Creates a routing info based on a text.

[RoutingInfo\(Object\)](#) - Constructor for class [routing.RoutingInfo](#)

Creates a routing info based on any object.

[RoutingInfoWindow](#) - Class in [gui](#)

A window for displaying routing information

[RoutingInfoWindow\(DTNHost\)](#) - Constructor for class [gui.RoutingInfoWindow](#)

[runSim\(\)](#) - Method in class [gui.DTNSimGUI](#)

[runSim\(\)](#) - Method in class [ui.DTNSimTextUI](#)

[runSim\(\)](#) - Method in class [ui.DTNSimUI](#)

Runs simulation after the model has been initialized.

S

[scale](#) - Static variable in class [gui.playfield.PlayFieldGraphic](#)

Common scaling factor for all playfield graphics.

[scale\(double\)](#) - Static method in class [gui.playfield.PlayFieldGraphic](#)

Scales the value according to current zoom level

[scale\(int\)](#) - Static method in class [gui.playfield.PlayFieldGraphic](#)

Scales the value according to current zoom level

[ScaleReferenceGraphic](#) - Class in [gui.playfield](#)

Reference scale bar graphic.

[ScaleReferenceGraphic\(\)](#) - Constructor for class [gui.playfield.ScaleReferenceGraphic](#)

[SCAN_ENERGY_S](#) - Static variable in class [routing.EnergyAwareRouter](#)

Energy usage per scanning -setting id ("scanEnergy").

[SCAN_INTERVAL_ID](#) - Static variable in class [core.NetworkInterface](#)

[ModuleCommunicationBus](#) identifier for the "scanning interval" variable.

[SCAN_INTERVAL_S](#) - Static variable in class [core.NetworkInterface](#)

scanning interval -setting id ("scanInterval")

[SCAN_INTERVAL_S](#) - Static variable in class [core.SimScenario](#)

scanning interval -setting id ("scanInterval")

[scen](#) - Variable in class [ui.DTNSimUI](#)

Scenario of the current simulation

[SCENARIO_NS](#) - Static variable in class [core.SimScenario](#)

namespace of scenario settings ("Scenario")

[**ScheduleDijkstra**](#) - Class in [routing.schedule](#)

Dijkstra's shortest path implementation for schedule data

[**ScheduleDijkstra\(ScheduleOracle\)**](#) - Constructor for class routing.schedule.[ScheduleDijkstra](#)

Constructor.

[**ScheduledUpdatesQueue**](#) - Class in [input](#)

Event queue where simulation objects can request an update to happen at the specified simulation time.

[**ScheduledUpdatesQueue\(\)**](#) - Constructor for class input.[ScheduledUpdatesQueue](#)

Constructor.

[**ScheduleEntry**](#) - Class in [routing.schedule](#)

[**ScheduleEntry\(double, int, int, int, double\)**](#) - Constructor for class routing.schedule.[ScheduleEntry](#)

Constructor of new schedule entry

[**ScheduleOracle**](#) - Class in [routing.schedule](#)

[**ScheduleOracle\(\)**](#) - Constructor for class routing.schedule.[ScheduleOracle](#)

[**scheduleUpdate\(double\)**](#) - Method in class core.[World](#)

Schedules an update request to all nodes to happen at the specified simulation time.

[**SECONDS_IN_UNIT_S**](#) - Static variable in class routing.[ProphetRouter](#)

Number of seconds in time unit -setting id ("secondsInTimeUnit").

[**selectDestination\(\)**](#) - Method in class movement.map.[PointsOfInterest](#)

Selects a random destination from POIs or all MapNodes.

[**selectRandomOkNode\(List<MapNode>\)**](#) - Method in class movement.[MapBasedMovement](#)

Selects and returns a random node that is OK from a list of nodes.

[**SEND**](#) - Static variable in class input.[StandardEventsReader](#)

Identifier of message transfer start event ("S")

[**SEND_QUEUE_MODE_S**](#) - Static variable in class routing.[MessageRouter](#)

Message/fragment sending queue type -setting id ("sendQueue").

[**sendEventToListeners\(String, Object, DTNHost\)**](#) - Method in class core.[Application](#)

Sends an event to all listeners.

[**SENDING**](#) - Static variable in class input.[MessageRelayEvent](#)

Message relay stage constant for start of sending

[**sendingConnections**](#) - Variable in class routing.[ActiveRouter](#)

connection(s) that are currently used for sending

[**sendMessage\(String, DTNHost\)**](#) - Method in class core.[DTNHost](#)

Sends a message from this host to another host

[**sendMessage\(String, DTNHost\)**](#) - Method in class routing.[MessageRouter](#)

Start sending a message to another host.

[**setAllHosts\(Collection<DTNHost>\)**](#) - Method in class report.[AdjacencyGraphvizReport](#)

Sets all hosts that should be in the graph at least once

[**setAlpha\(double\)**](#) - Method in class routing.maxprop.[MeetingProbabilitySet](#)

Enables changing the alpha parameter dynamically

[**setAppID\(String\)**](#) - Method in class core.[Application](#)

Sets the application ID.

[**setAppID\(String\)**](#) - Method in class core.[Message](#)

[**setAppListeners\(List<ApplicationListener>\)**](#) - Method in class core.[Application](#)

[**setAutoClearOverlay\(boolean\)**](#) - Method in class gui.playfield.[PlayField](#)

Enables or disables the automatic clearing of overlay graphics.

[**setBidirectional\(boolean\)**](#) - Method in class input.[WKTMapReader](#)

Sets bidirectional paths on/off.

[**setBusStops\(List<Coord>\)**](#) - Method in class movement.[BusControlSystem](#)

Set the bus stops that belong to this system

[**setListeners\(List<ConnectionListener>\)**](#) - Method in class core.[NetworkInterface](#)

For setting the connectionListeners

[**setComBus\(ModuleCommunicationBus\)**](#) - Method in class movement.[MovementModel](#)

Sets the module communication bus for this movement model

[**setCurrentMovementModel\(SwitchableMovement\)**](#) - Method in class movement.[ExtendedMovementModel](#)

Sets the current movement model to be used the next time getPath() is called

[**setDebugLevel\(int\)**](#) - Method in class core.[Debug](#)

Sets the current debug level (smaller level -> more messages)

[**setDelta\(double\)**](#) - Method in class routing.schedule.[ScheduleEntry](#)

[**setDestination\(Coord\)**](#) - Method in class movement.[EveningTrip](#)

Sets the destination square of the trip.

[**setDestMax\(int\)**](#) - Method in class applications.[PingApplication](#)

[**setDestMin\(int\)**](#) - Method in class applications.[PingApplication](#)

[**setDone\(boolean\)**](#) - Method in class input.[WKTReader](#)

Sets the "is file read" state

[**setDrawConnections\(boolean\)**](#) - Static method in class gui.playfield.[NodeGraphic](#)

Sets whether node's connections to other nodes should be drawn

[**setDrawCoverage\(boolean\)**](#) - Static method in class gui.playfield.[NodeGraphic](#)

Sets whether radio coverage of nodes should be drawn

[**setDrawnodeName\(boolean\)**](#) - Static method in class gui.playfield.[NodeGraphic](#)

Sets whether node's name should be displayed

[**setEncounters\(int\[\]\)**](#) - Method in class report.[TotalEncountersReport](#)

[**setEnergy\(double\[\]\)**](#) - Method in class routing.[EnergyAwareRouter](#)

Sets the current energy level into the given range using uniform random distribution.

[**setEvents\(DTN2Events\)**](#) - Static method in class core.[DTN2Manager](#)

Sets the DTN2Events object.

[**setFocus\(DTNHost\)**](#) - Method in class gui.[DTNSimGUI](#)

Sets a node's graphical presentation in the center of the playfield view

[**setHost\(DTNHost\)**](#) - Method in class core.[NetworkInterface](#)

For setting the host - needed when a prototype is copied for several hosts

[**setInterval\(double\)**](#) - Method in class applications.[PingApplication](#)

[**setLastPing\(double\)**](#) - Method in class applications.[PingApplication](#)

[**setLocation\(double, double\)**](#) - Method in class core.[Coord](#)

Sets the location of this coordinate object

[**setLocation\(Coord\)**](#) - Method in class core.[Coord](#)

Sets this coordinate's location to be equal to other coordinates location

[**setLocation\(Coord\)**](#) - Method in class core.[DTNHost](#)

Sets the Node's location overriding any location set by movement model

[**setLocation\(Coord\)**](#) - Method in class movement.[BusTravellerMovement](#)

[**setLocation\(Coord\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**setLocation\(Coord\)**](#) - Method in class movement.[HomeActivityMovement](#)

[**setLocation\(Coord\)**](#) - Method in class movement.[MapBasedMovement](#)

[**setLocation\(Coord\)**](#) - Method in class movement.[OfficeActivityMovement](#)

[**setLocation\(Coord\)**](#) - Method in class movement.[RandomWalk](#)

[**setLocation\(Coord\)**](#) - Method in interface movement.[SwitchableMovement](#)

Tell the movement model what its current location is

[**setMap\(SimMap\)**](#) - Method in class gui.playfield.[PlayField](#)

Sets the source for the map graphics and enables map graphics showing

[**setMap\(SimMap\)**](#) - Method in class movement.[BusControlSystem](#)

Provide the system with the map

[**setMaxGroupSize\(int\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**setMeetingSpots\(List<Coord>\)**](#) - Method in class movement.[EveningActivityControlSystem](#)

Sets the meeting locations the nodes can choose among

[**setMinGroupSize\(int\)**](#) - Method in class movement.[EveningActivityMovement](#)

[**setName\(String\)**](#) - Method in class core.[DTNHost](#)

Sets the Node's name overriding the default name (groupId + netAddress)

[**setNameSpace\(String\)**](#) - Method in class core.[Settings](#)

Sets the namespace to something else than the current namespace.

[**setNextEventQueue\(\)**](#) - Method in class core.[World](#)

Goes through all event Queues and sets the event queue that has the next event.

[**setNextIndex\(int\)**](#) - Method in class movement.map.[MapRoute](#)

Sets the next index for this route

[**setNextRoute\(Coord, Coord\)**](#) - Method in class movement.[BusTravellerMovement](#)

Sets the next route for the traveller, so that it can decide whether it should take the bus or not.

[**setNextRoute\(Coord, Coord\)**](#) - Method in class movement.[CarMovement](#)

Sets the next route to be taken

[**setNextRoute\(Coord, Coord\)**](#) - Method in interface movement.[TransportMovement](#)

[**setNodeRelationships\(int\[\]\[\]\)**](#) - Method in class report.[UniqueEncountersReport](#)

[**setNormalize\(boolean\)**](#) - Method in class input.[ExternalMovementReader](#)

Sets normalizing of read values on/off.

[**setNrofPreload\(int\)**](#) - Method in class input.[ExternalEventsQueue](#)

Sets maximum number of events that are read when the next preload occurs

[**setPassive\(boolean\)**](#) - Method in class applications.[PingApplication](#)

[**setPath\(Path\)**](#) - Method in class movement.[EveningTrip](#)

Sets the shopping path for the group

[**setPaused\(boolean\)**](#) - Method in class gui.[DTNSimGUI](#)

Sets the pause of the simulation on/off

[**setPaused\(boolean\)**](#) - Method in class gui.[GUIControls](#)

Sets simulation to pause or play.

[**setPauseOnEvent\(boolean\)**](#) - Method in class gui.[EventLogControl](#)

Sets ought this event type cause pause (return true for [EventLogControl.pauseOnEvent\(\)](#))

[**setPingSize\(int\)**](#) - Method in class applications.[PingApplication](#)

[**setPongSize\(int\)**](#) - Method in class applications.[PingApplication](#)

[**setPrefix\(String\)**](#) - Method in class report.[Report](#)

Sets a prefix that will be inserted before every line in the report

[**setPrintStream\(PrintStream\)**](#) - Method in class core.[Debug](#)

Sets print stream of debug output.

[**setRandomNumberGenerator\(Random\)**](#) - Method in class movement.[EveningActivityControlSystem](#)

Sets the random number generator to be used

[**setReceiveTime\(double\)**](#) - Method in class core.[Message](#)

Sets the time when this message was received.

[**setReporter\(DTN2Reporter\)**](#) - Static method in class core.[DTN2Manager](#)

Sets the DTN2Reporter object used to pass messages from ONE to dtnd.

[**setRequest\(Message\)**](#) - Method in class core.[Message](#)

If this message is a response to a request, sets the request message

[**setResponseSize\(int\)**](#) - Method in class core.[Message](#)

Sets the requested response message's size.

[**setRunIndex\(int\)**](#) - Static method in class core.[Settings](#)

Sets the run index for the settings (only has effect on settings with run array).

[setScale\(double\)](#) - Method in class gui.playfield.[PlayField](#)

Sets the zooming/scaling factor

[setScale\(double\)](#) - Static method in class gui.playfield.[PlayFieldGraphic](#)

Set the zooming factor of the graphics to be drawn

[setSecondaryNamespace\(String\)](#) - Method in class core.[Settings](#)

Sets a secondary namespace where a setting is searched from if it isn't found from the primary namespace.

[setSeed\(int\)](#) - Method in class applications.[PingApplication](#)

[setShowEvent\(boolean\)](#) - Method in class gui.[EventLogControl](#)

Sets ought this event type should be shown (return true for [EventLogControl.showEvent\(\)](#))

[setShowMapGraphic\(boolean\)](#) - Method in class gui.playfield.[PlayField](#)

Enables/disables showing of map graphics

[setSimTime\(double\)](#) - Method in class gui.[GUIControls](#)

Sets the simulation time that control panel shows

[setSpeed\(double\)](#) - Method in class movement.[Path](#)

Sets a constant speed for the whole path.

[setTime\(double\)](#) - Method in class core.[SimClock](#)

Sets the time of the clock.

[SETTING_OUTPUT_S](#) - Static variable in class core.[Settings](#)

Setting to define the file name where all read settings are written ("Settings.output").

[Settings](#) - Class in [core](#)

Interface for simulation settings stored in setting file(s).

[Settings\(String\)](#) - Constructor for class core.[Settings](#)

Creates a setting object with a namespace.

[Settings\(\)](#) - Constructor for class core.[Settings](#)

Create a setting object without namespace.

[SETTINGS_NAMESPACE](#) - Static variable in class input.[EventQueueHandler](#)

Event queue settings main namespace ("Events")

[SETTINGS_NAMESPACE](#) - Static variable in class input.[ExternalEventsQueue](#)

ExternalEvents namespace ("ExternalEvents")

[SETTINGS_NS](#) - Static variable in class core.[World](#)

namespace of optimization settings ("Optimization")

[SettingsError](#) - Error in [core](#)

Settings related error

[SettingsError\(String\)](#) - Constructor for error core.[SettingsError](#)

[SettingsError\(String, Exception\)](#) - Constructor for error core.[SettingsError](#)

[SettingsError\(Exception\)](#) - Constructor for error core.[SettingsError](#)

[setTtl\(int\)](#) - Method in class core.[Message](#)

Sets the initial TTL (time-to-live) for this message.

[setUnderlayImage\(BufferedImage, double, double, double, double\)](#) - Method in class gui.playfield.[PlayField](#)

Sets an image to show under the host graphics

[setup\(World\)](#) - Static method in class core.[DTN2Manager](#)

Sets up the dtnd connections by parsing the configuration file defined in the DTN2.configFile setting.

[setUpState\(boolean\)](#) - Method in class core.[Connection](#)

Sets the state of the connection.

[setWaitTimeAtEnd\(double\)](#) - Method in class movement.[EveningTrip](#)

[ShortestPathMapBasedMovement](#) - Class in [movement](#)

Map based movement model that uses Dijkstra's algorithm to find shortest paths between two random map nodes and Points Of Interest

[ShortestPathMapBasedMovement\(Settings\)](#) - Constructor for class movement.[ShortestPathMapBasedMovement](#)

Creates a new movement model based on a Settings object's settings.

[ShortestPathMapBasedMovement\(ShortestPathMapBasedMovement\)](#) - Constructor for class

[movement](#).[ShortestPathMapBasedMovement](#)

Copyconstructor.

[showEvent\(\)](#) - Method in class [gui](#).[EventLogControl](#)

Returns true if this event type should be shown

[showInfo\(DTNHost\)](#) - Method in class [gui](#).[InfoPanel](#)

Show information about a host

[showInfo\(Message\)](#) - Method in class [gui](#).[InfoPanel](#)

Show information about a message

[showPath\(Path\)](#) - Method in class [gui](#).[DTNSimGUI](#)

Shows a path on the playfield

[shuffleMessages\(List<Message>\)](#) - Method in class [routing](#).[ActiveRouter](#)

Shuffles a messages list so the messages are in random order.

[SIM_CON_S](#) - Static variable in class [core](#).[SimScenario](#)

simulate connections -setting id ("simulateConnections")

[simCancelled](#) - Variable in class [ui](#).[DTNSimUI](#)

is simulation termination requested

[SimClock](#) - Class in [core](#)

Wall clock for checking the simulation time.

[simDone](#) - Variable in class [ui](#).[DTNSimUI](#)

has simulation terminated normally

[SimError](#) - Error in [core](#)

Error in the simulation

[SimError\(String\)](#) - Constructor for error [core](#).[SimError](#)

[SimError\(String, Exception\)](#) - Constructor for error [core](#).[SimError](#)

[SimError\(Exception\)](#) - Constructor for error [core](#).[SimError](#)

[SimMap](#) - Class in [movement.map](#)

A simulation map for node movement.

[SimMap\(Map<Coord, MapNode>\)](#) - Constructor for class [movement.map](#).[SimMap](#)

[SimMenuBar](#) - Class in [gui](#)

Menu bar of the simulator GUI

[SimMenuBar\(PlayField\)](#) - Constructor for class [gui](#).[SimMenuBar](#)

[SimpleBroadcastInterface](#) - Class in [interfaces](#)

A simple Network Interface that provides a constant bit-rate service, where one transmission can be on at a time.

[SimpleBroadcastInterface\(Settings\)](#) - Constructor for class [interfaces](#).[SimpleBroadcastInterface](#)

Reads the interface settings from the Settings file

[SimpleBroadcastInterface\(SimpleBroadcastInterface\)](#) - Constructor for class

[interfaces](#).[SimpleBroadcastInterface](#)

Copy constructor

[SimScenario](#) - Class in [core](#)

A simulation scenario used for getting and storing the settings of a simulation run.

[SimScenario\(\)](#) - Constructor for class [core](#).[SimScenario](#)

Creates a scenario based on Settings object.

[simulateConnections\(\)](#) - Method in class [core](#).[SimScenario](#)

Returns true if connections should be simulated

[skipAllWhitespace\(Reader\)](#) - Method in class [input](#).[WKTReader](#)

Skips all consecutive whitespace characters from reader

[skipUntil\(Reader, char\)](#) - Method in class [input](#).[WKTReader](#)

Reads and skips all characters until character "until" is read or end of stream is reached.

[sortByQueueMode\(List\)](#) - Method in class [routing](#).[MessageRouter](#)

Sorts/shuffles the given list according to the current sending queue mode.

[SPEED](#) - Static variable in class [movement](#).[MovementModel](#)

node's speed CSV (min, max) -setting id ("speed")

SPEED_ID - Static variable in class core.[NetworkInterface](#)

[ModuleCommunicationBus](#) identifier for the "transmission speed" variable.

SPRAYANDWAIT_NS - Static variable in class routing.[SprayAndWaitRouter](#)

SprayAndWait router's settings name space ("SprayAndWaitRouter")

SprayAndWaitRouter - Class in [routing](#)

Implementation of Spray and wait router as depicted in *Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks* by Thrasivoulos Spyropoulos et al.

SprayAndWaitRouter(Settings) - Constructor for class routing [SprayAndWaitRouter](#)

SprayAndWaitRouter(SprayAndWaitRouter) - Constructor for class routing [SprayAndWaitRouter](#)

Copy constructor.

STAGE_STRINGS - Static variable in class input.[MessageRelayEvent](#)

Stage constant -> String representation mapping

StandardEventsReader - Class in [input](#)

External events reader for standard-format events (created e.g by the dtnsim2parser).

StandardEventsReader(File) - Constructor for class input [StandardEventsReader](#)

start() - Method in class ui.[DTNSimUI](#)

Starts the simulation.

START_LOCATION_S - Static variable in class movement.[LinearFormation](#)

Per node group setting for defining the start coordinates of the line ("startLocation")

startTiming(String) - Static method in class core.[Debug](#)

Start timing an action.

startTransfer(DTNHost, Message) - Method in class core.[CBRConnection](#)

Sets a message that this connection is currently transferring.

startTransfer(DTNHost, Message) - Method in class core.[Connection](#)

Sets a message that this connection is currently transferring.

startTransfer(DTNHost, Message) - Method in class core.[VBRConnection](#)

Sets a message that this connection is currently transferring.

startTransfer(Message, Connection) - Method in class routing.[ActiveRouter](#)

Tries to start a transfer of message using a connection.

STATE_DECIDED_TO_ENTER_A_BUS - Static variable in class movement.[BusTravellerMovement](#)

STATE_TRAVELLING_ON_BUS - Static variable in class movement.[BusTravellerMovement](#)

STATE_WAITING_FOR_BUS - Static variable in class movement.[BusTravellerMovement](#)

STATE_WALKING_ELSEWHERE - Static variable in class movement.[BusTravellerMovement](#)

stateChanged(ChangeEvent) - Method in class gui.[GUIControls](#)

StationaryMovement - Class in [movement](#)

A dummy stationary "movement" model where nodes do not move.

StationaryMovement(Settings) - Constructor for class movement.[StationaryMovement](#)

Creates a new movement model based on a Settings object's settings.

StationaryMovement(StationaryMovement) - Constructor for class movement.[StationaryMovement](#)

Copy constructor.

STD_FOR_TIME_DIFF_SETTING - Static variable in class movement.[HomeActivityMovement](#)

storeToBinaryFile(String, List<ExternalEvent>) - Static method in class input.[BinaryEventsReader](#)

Stores the events to a binary file

subscribe(String, ModuleCommunicationListener) - Method in class core.[ModuleCommunicationBus](#)

Subscribes a module to changes of a certain value.

SwitchableMovement - Interface in [movement](#)

Movement models to be used by ExtendedMovementModels should implement this interface

SYNTAX - Static variable in class report.[DistanceDelayReport](#)

Syntax of the report lines

T

[**time**](#) - Variable in class input.[ExternalEvent](#)

Time of the event (simulated seconds)

[**TIME SCALE S**](#) - Static variable in class routing.[MaxPropRouterWithEstimation](#)

[**TIME SCALE S**](#) - Static variable in class routing.[ProphetRouterWithEstimation](#)

Number of seconds in time scale.

[**TO HOST RANGE S**](#) - Static variable in class input.[MessageEventGenerator](#)

(Optional) receiver address range -setting id ("tohosts").

[**toAddr**](#) - Variable in class input.[ConnectionEvent](#)

address of the node the (dis)connection is to

[**toAddr**](#) - Variable in class input.[MessageEvent](#)

address of the node the message is to

[**toHostRange**](#) - Variable in class input.[MessageEventGenerator](#)

Range of host addresses that can be receivers

[**toInterface**](#) - Variable in class core.[Connection](#)

[**toNode**](#) - Variable in class core.[Connection](#)

[**toString\(\)**](#) - Method in class core.[CBRConnection](#)

Returns a String presentation of the connection.

[**toString\(\)**](#) - Method in class core.[Connection](#)

Returns a String presentation of the connection.

[**toString\(\)**](#) - Method in class core.[Coord](#)

Returns a text representation of the coordinate (rounded to 2 decimals)

[**toString\(\)**](#) - Method in class core.[DTNHost](#)

Returns a string presentation of the host.

[**toString\(\)**](#) - Method in class core.[Message](#)

Returns a string representation of the message

[**toString\(\)**](#) - Method in class core.[ModuleCommunicationBus](#)

[**toString\(\)**](#) - Method in class core.[NetworkInterface](#)

Returns a string representation of the object.

[**toString\(\)**](#) - Method in class core.[Settings](#)

Returns a String representation of the stored settings

[**toString\(\)**](#) - Method in class core.[SimClock](#)

Returns the current simulation time in a string

[**toString\(\)**](#) - Method in class core.[Tuple](#)

Returns a string representation of the tuple

[**toString\(\)**](#) - Method in class core.[VBRConnection](#)

Returns a String presentation of the connection.

[**toString\(\)**](#) - Method in class gui.[EventLogPanel](#)

[**toString\(\)**](#) - Method in class input.[ConnectionEvent](#)

[**toString\(\)**](#) - Method in class input.[ExternalEvent](#)

Returns a String representation of the event

[**toString\(\)**](#) - Method in class input.[MessageCreateEvent](#)

[**toString\(\)**](#) - Method in class input.[MessageDeleteEvent](#)

[**toString\(\)**](#) - Method in class input.[MessageEvent](#)

[toString\(\)](#) - Method in class input.[MessageRelayEvent](#)

[toString\(\)](#) - Method in class input.[ScheduledUpdatesQueue](#)

[toString\(\)](#) - Method in class interfaces.[ConnectivityGrid.GridCell](#)

Returns a string representation of the cell

[toString\(\)](#) - Method in class interfaces.[ConnectivityGrid](#)

Returns a string representation of the ConnectivityCells object

[toString\(\)](#) - Method in class interfaces.[InterferenceLimitedInterface](#)

Returns a string representation of the object.

[toString\(\)](#) - Method in class interfaces.[SimpleBroadcastInterface](#)

Returns a string representation of the object.

[toString\(\)](#) - Method in class movement.map.[MapNode](#)

Returns a String representation of the map node

[toString\(\)](#) - Method in class movement.map.[MapRoute](#)

[toString\(\)](#) - Method in class movement.map.[SimMap](#)

Returns a String representation of the map

[toString\(\)](#) - Method in class movement.[MovementModel](#)

Returns simply the name of the movement model class

[toString\(\)](#) - Method in class movement.[Path](#)

Returns a string presentation of the path's coordinates

[toString\(\)](#) - Method in class report.[ContactTimesReport.ConnectionInfo](#)

Returns a string representation of the info object

[toString\(\)](#) - Method in class routing.[EnergyAwareRouter](#)

[toString\(\)](#) - Method in class routing.maxprop.[MeetingProbabilitySet](#)

Returns a String presentation of the probabilities

[toString\(\)](#) - Method in class routing.[MessageRouter](#)

Returns a String presentation of this router

[toString\(\)](#) - Method in class routing.[RoutingInfo](#)

Returns the info text of this routing info.

[toString\(\)](#) - Method in class routing.schedule.[ScheduleEntry](#)

[TotalContactTimeReport](#) - Class in [report](#)

Report for total amount of contact times among hosts.

[TotalContactTimeReport\(\)](#) - Constructor for class report.[TotalContactTimeReport](#)

[TotalEncountersReport](#) - Class in [report](#)

A report of the distribution of how many encounters (contacts) a node has had

[TotalEncountersReport\(\)](#) - Constructor for class report.[TotalEncountersReport](#)

[transferAborted\(Connection\)](#) - Method in class routing.[ActiveRouter](#)

Method is called just before a transfer is aborted at [ActiveRouter.update\(\)](#) due connection going down.

[transferDone\(Connection\)](#) - Method in class routing.[ActiveRouter](#)

Method is called just before a transfer is finalized at [ActiveRouter.update\(\)](#).

[transferDone\(Connection\)](#) - Method in class routing.[EpidemicOracleRouter](#)

[transferDone\(Connection\)](#) - Method in class routing.[FirstContactRouter](#)

[transferDone\(Connection\)](#) - Method in class routing.[MaxPropRouter](#)

Method is called just before a transfer is finalized at [ActiveRouter.update\(\)](#).

[transferDone\(Connection\)](#) - Method in class routing.[MaxPropRouterWithEstimation](#)

Method is called just before a transfer is finalized at [ActiveRouter.update\(\)](#).

[transferDone\(Connection\)](#) - Method in class routing.[SprayAndWaitRouter](#)

Called just before a transfer is finalized (by [ActiveRouter.update\(\)](#)).

[TRANSFERRED](#) - Static variable in class input.[MessageRelayEvent](#)

Message relay stage constant for ready delivery

[**translate\(double, double\)**](#) - Method in class core.[Coord](#)

Moves the point by dx and dy

[**translate\(double, double\)**](#) - Method in class movement.map.[SimMap](#)

Translate whole map by dx and dy

[**TRANSMIT ENERGY S**](#) - Static variable in class routing.[EnergyAwareRouter](#)

Energy usage per second when sending -setting id ("transmitEnergy").

[**TRANSMIT RANGE S**](#) - Static variable in class core.[NetworkInterface](#)

transmit range -setting id ("transmitRange")

[**TRANSMIT SPEED S**](#) - Static variable in class core.[NetworkInterface](#)

transmit speed -setting id ("transmitSpeed")

[**transmitRange**](#) - Variable in class core.[NetworkInterface](#)

[**transmitSpeed**](#) - Variable in class core.[NetworkInterface](#)

[**TransportMovement**](#) - Interface in [movement](#)

MovementModels used for transportation should implement this interface

[**TRY LATER BUSY**](#) - Static variable in class routing.[MessageRouter](#)

Receive return value for busy receiver

[**tryAllMessages\(Connection, List<Message>\)**](#) - Method in class routing.[ActiveRouter](#)

Goes trough the messages until the other node accepts one for receiving (or doesn't accept any).

[**tryAllMessagesToAllConnections\(\)**](#) - Method in class routing.[ActiveRouter](#)

Tries to send all messages that this router is carrying to all connections this node has.

[**tryMessagesForConnected\(List<Tuple<Message, Connection>>\)**](#) - Method in class routing.[ActiveRouter](#)

Tries to send messages for the connections that are mentioned in the Tuples in the order they are in the list until one of the connections starts transferring or all tuples have been tried.

[**tryMessagesToConnections\(List<Message>, List<Connection>\)**](#) - Method in class routing.[ActiveRouter](#)

Tries to send all given messages to all given connections.

[**TTL CHECK INTERVAL**](#) - Static variable in class routing.[ActiveRouter](#)

how often TTL check (discarding old messages) is performed

[**Tuple<K,V>**](#) - Class in [core](#)

A generic key-value tuple.

[**Tuple\(K, V\)**](#) - Constructor for class core.[Tuple](#)

Creates a new tuple.

U

[**ui**](#) - package ui

Contains superclass for all user interfaces and a simple user interface(s).

[**UI_UP_INTERVAL**](#) - Static variable in class ui.[DTNSimTextUI](#)

How often the UI view is updated (milliseconds)

[**UniqueEncountersReport**](#) - Class in [report](#)

UniqueEncountersReport class creates a report of the distribution of how many promilles of the other nodes a node has encountered.

[**UniqueEncountersReport\(\)**](#) - Constructor for class report.[UniqueEncountersReport](#)

[**unsubscribe\(String, ModuleCommunicationListener\)**](#) - Method in class core.[ModuleCommunicationBus](#)

Removes a notification subscription

[**UP_INT_S**](#) - Static variable in class core.[SimScenario](#)

update interval -setting id ("updateInterval")

[**UP_SPEEDS**](#) - Static variable in class gui.[GUIControls](#)

GUI update speeds.

[**update\(DTNHost\)**](#) - Method in class applications.[PingApplication](#)

Sends a ping packet if this is an active application instance.

[**update\(DTNHost\)**](#) - Method in class core.[Application](#)

Called every simulation cycle.

[update\(\)](#) - Method in class core.[Connection](#)

Calculate the current transmission speed from the information given by the interfaces, and calculate the missing data amount.

[update\(boolean\)](#) - Method in class core.[DTNHost](#)

Updates node's network layer and router.

[update\(\)](#) - Method in class core.[NetworkInterface](#)

Updates the state of current connections (ie tears down connections that are out of range, recalculates transmission speeds etc.).

[update\(\)](#) - Method in class core.[VBRCConnection](#)

Calculate the current transmission speed from the information given by the interfaces, and calculate the missing data amount.

[update\(\)](#) - Method in class core.[World](#)

Update (move, connect, disconnect etc.) all hosts in the world.

[update\(boolean\)](#) - Method in class gui.[DTNSimGUI](#)

Updates the GUI

[update\(\)](#) - Method in class interfaces.[InterferenceLimitedInterface](#)

Updates the state of current connections (i.e., tears down connections that are out of range).

[update\(\)](#) - Method in class interfaces.[SimpleBroadcastInterface](#)

Updates the state of current connections (ie tears down connections that are out of range).

[update\(\)](#) - Method in class routing.[ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

[update\(\)](#) - Method in class routing.[DirectDeliveryRouter](#)[update\(\)](#) - Method in class routing.[EnergyAwareRouter](#)[update\(\)](#) - Method in class routing.[EpidemicOracleRouter](#)[update\(\)](#) - Method in class routing.[EpidemicRouter](#)[update\(\)](#) - Method in class routing.[FirstContactRouter](#)[update\(\)](#) - Method in class routing.[MaxPropRouter](#)[update\(\)](#) - Method in class routing.[MaxPropRouterWithEstimation](#)[update\(\)](#) - Method in class routing.[MessageRouter](#)

Updates router.

[update\(\)](#) - Method in class routing.[PassiveRouter](#)[update\(\)](#) - Method in class routing.[ProphetRouter](#)[update\(\)](#) - Method in class routing.[ProphetRouterWithEstimation](#)[update\(\)](#) - Method in class routing.[SprayAndWaitRouter](#)[updated\(List<DTNHost>\)](#) - Method in interface core.[UpdateListener](#)

Method is called on every update cycle.

[updated\(List<DTNHost>\)](#) - Method in class report.[ContactsDuringAnICTReport](#)[updated\(List<DTNHost>\)](#) - Method in class report.[EncountersVSUniqueEncountersReport](#)[updated\(List<DTNHost>\)](#) - Method in class report.[EnergyLevelReport](#)

Creates a new snapshot of the energy levels if "granularity" seconds have passed since the last snapshot.

[updated\(List<DTNHost>\)](#) - Method in class report.[MessageLocationReport](#)

Creates a new snapshot of the message locations if "granularity" seconds have passed since the last snapshot.

[updated\(List<DTNHost>\)](#) - Method in class report.[TotalContactTimeReport](#)

Reports total contact time if more time than defined with setting [ContactTimesReport.GRANULARITY](#) has

passed.

[updated\(List<DTNHost>\)](#) - Method in class report.[TotalEncountersReport](#)

[updated\(List<DTNHost>\)](#) - Method in class report.[UniqueEncountersReport](#)

[updateDouble\(String, double\)](#) - Method in class core.[ModuleCommunicationBus](#)

Changes the Double value with given key with the value delta

[updateEstimators\(DTNHost\)](#) - Method in class routing.[MaxPropRouterWithEstimation](#)

Updates the MaxPROP estimators

[updateField\(\)](#) - Method in class gui.playfield.[PlayField](#)

Schedule the play field to be drawn

[UpdateListener](#) - Interface in [core](#)

Interface for classes that want to be informed about every single update call to the World object.

[updateLocation\(NetworkInterface\)](#) - Method in class interfaces.[ConnectivityGrid](#)

Checks and updates (if necessary) interface's position in the grid

[updateLocation\(NetworkInterface\)](#) - Method in class interfaces.[ConnectivityOptimizer](#)

Updates a network interface's location

[updateMeetingProbFor\(Integer\)](#) - Method in class routing.maxprop.[MeetingProbabilitySet](#)

Updates meeting probability for the given node index.

[updateMeetingProbFor\(Integer, double\)](#) - Method in class routing.maxprop.[MeetingProbabilitySet](#)

[updateParam\(\)](#) - Method in class routing.[MaxPropRouterWithEstimation](#)

update the alpha parameter based on the estimators

[updateProperty\(String, Object\)](#) - Method in class core.[Message](#)

Updates a value for an existing property.

[updateProperty\(String, Object\)](#) - Method in class core.[ModuleCommunicationBus](#)

Updates a value for an existing property.

V

[valueFillString\(String\)](#) - Method in class core.[Settings](#)

Fills a String formatted in a special way with values from Settings.

[VBRConnection](#) - Class in [core](#)

A connection between two DTN nodes.

[VBRConnection\(DTNHost, NetworkInterface, DTNHost, NetworkInterface\)](#) - Constructor for class core.[VBRConnection](#)

Creates a new connection between nodes and sets the connection state to "up".

W

[WAIT_TIME](#) - Static variable in class movement.[MovementModel](#)

node's wait time CSV (min, max) -setting id ("waitTime")

[WARMUP_S](#) - Static variable in class report.[Report](#)

Warm up period -setting id ("warmup").

[WARMUP_S](#) - Static variable in class routing.[EnergyAwareRouter](#)

Energy update warmup period -setting id ("energyWarmup").

[warmupIDs](#) - Variable in class report.[Report](#)

[warmupMovementModel\(double\)](#) - Method in class core.[World](#)

Moves hosts in the world for the time given time initialize host positions properly.

[warmupTime](#) - Variable in class report.[Report](#)

[WKTMapReader](#) - Class in [input](#)

"Well-known text syntax" map data reader.

Note: Understands only LINESTRINGS and MULTILINESTRINGS.

[**WKTMapReader\(boolean\)**](#) - Constructor for class input.[**WKTMapReader**](#)

Constructor.

[**WKTReader**](#) - Class in [input](#)

Class for reading "Well-known text syntax" files.

[**WKTReader\(\)**](#) - Constructor for class input.[**WKTReader**](#)

[**WORK_DAY_LENGTH_SETTING**](#) - Static variable in class movement.[**OfficeActivityMovement**](#)

[**WorkingDayMovement**](#) - Class in [movement](#)

This movement model makes use of several other movement models to simulate movement with daily routines.

[**WorkingDayMovement\(Settings\)**](#) - Constructor for class movement.[**WorkingDayMovement**](#)

Creates a new instance of WorkingDayMovement

[**WorkingDayMovement\(WorkingDayMovement\)**](#) - Constructor for class movement.[**WorkingDayMovement**](#)

Creates a new instance of WorkingDayMovement from a prototype

[**World**](#) - Class in [core](#)

World contains all the nodes and is responsible for updating their location and connections.

[**World\(List<DTNHost>, int, int, double, List<UpdateListener>, boolean, List<EventQueue>\)**](#) - Constructor for class core.[**World**](#)

Constructor.

[**world**](#) - Variable in class ui.[**DTNSimUI**](#)

The World where all actors of the simulator are

[**WORLD_SIZE**](#) - Static variable in class movement.[**MovementModel**](#)

world's size CSV (width, height) -setting id ("worldSize")

[**write\(String\)**](#) - Method in class report.[**Report**](#)

Writes a line to report using defined prefix and [**Report.out**](#) writer.

Z

[**ZOOM_MAX**](#) - Static variable in class gui.[**GUIControls**](#)

Highest value for the zoom level

[**ZOOM_MIN**](#) - Static variable in class gui.[**GUIControls**](#)

Smallest value for the zoom level

[**A**](#) [**B**](#) [**C**](#) [**D**](#) [**E**](#) [**F**](#) [**G**](#) [**H**](#) [**I**](#) [**L**](#) [**M**](#) [**N**](#) [**O**](#) [**P**](#) [**Q**](#) [**R**](#) [**S**](#) [**T**](#) [**U**](#) [**V**](#) [**W**](#) [**Z**](#)

[**Overview**](#) [**Package**](#) [**Class**](#) [**Tree**](#) [**Deprecated**](#) [**Index**](#) [**Help**](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Overview

The [Overview](#) page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain four categories:

- Interfaces (*italic*)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description
- Nested Class Summary
- Field Summary
- Constructor Summary
- Method Summary
- Field Detail
- Constructor Detail
- Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type declaration
- Annotation Type description
- Required Element Summary
- Optional Element Summary
- Element Detail

Enum

Each enum has its own separate page with the following sections:

- Enum declaration
- Enum description
- Enum Constant Summary
- Enum Constant Detail

Tree (Class Hierarchy)

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

Deprecated API

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The [Index](#) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

Prev/Next

These links take you to the next or previous class, interface, package, or related page.

Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

Constant Field Values

The [Constant Field Values](#) page lists the static final fields and their values.

This help file applies to API documentation generated using the standard doclet.

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV NEXT

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Packages

applications	
core	Contains core classes and interfaces of the simulator.
gui	Contains the classes of Graphical User Interface.
gui.playfield	Contains the classes of Graphical User Interface's playfield -view (the graphical presentation of the nodes' locations and other information).
input	Provides interfaces and classes for reading input data from external sources.
interfaces	
movement	Contains different movement models and related classes for the simulator.
movement.map	Sub package for MapBasedMovement movement model's (and its sub classes) helper classes.
report	Contains all the report classes.
routing	Contains all the router classes who decide how to handle the messages.
routing.maxprop	Contains MaxProp routing module specific classes.
routing.schedule	
ui	Contains superclass for all user interfaces and a simple user interface(s).

Package applications

Class Summary

PingApplication	Simple ping application to demonstrate the application support.
---------------------------------	---

Package core

Contains core classes and interfaces of the simulator.

See:

[Description](#)

Interface Summary

ApplicationListener	Interface for classes that want to be informed about messages between hosts.
ConnectionListener	Interface for classes that want to be informed about connections between hosts.
MessageListener	Interface for classes that want to be informed about messages between hosts
ModuleCommunicationListener	This interface should be implemented by classes that want to be notified of variable value changes in ModuleCommunicationBuses.
MovementListener	Interface for classes that want to be informed about node movement.
UpdateListener	Interface for classes that want to be informed about every single update call to the World object.

Class Summary

Application	Base class for applications.
CBRConnection	A constant bit-rate connection between two DTN nodes.
Connection	A connection between two DTN nodes.
Coord	Class to hold 2D coordinates and perform simple arithmetics and transformations
Debug	Debugging info printer with time stamping.
DTN2Manager	Manages the external convergence layer connections to dtnd.
DTN2Manager.EIDHost	EID to DTNHost mapping elements.
DTNHost	A DTN capable host.
DTNSim	Simulator's main class
Message	A message that is created at a node or passed between nodes.
ModuleCommunicationBus	Intermodule communication bus.
NetworkInterface	Network interface of a DTNHost.
ParetoRNG	A random number generator for a Pareto distribution
Settings	Interface for simulation settings stored in setting file(s).
SimClock	Wall clock for checking the simulation time.
SimScenario	A simulation scenario used for getting and storing the settings of a simulation run.
Tuple<K,V>	A generic key-value tuple.
VBRConnection	A connection between two DTN nodes.
World	World contains all the nodes and is responsible for updating their location and

core

connections.

Error Summary

SettingsError	Settings related error
SimError	Error in the simulation

Package core Description

Contains core classes and interfaces of the simulator. Almost all of these classes are needed for every run of the simulator. DTNSim is the main class of the program. It starts up a proper user interface which in turn starts the simulation.

See Also:

[DTNSimUI](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Package gui

Contains the classes of Graphical User Interface.

See:

[Description](#)

Class Summary

DTNSimGUI	Graphical User Interface for simulator
EventLogControl	Class encapsulates the references to the controls one can add to the EventLogControlPanel
EventLogControlPanel	Control panel for event log
EventLogPanel	Event log panel where log entries are displayed.
GUIControls	GUI's control panel
InfoPanel	Information panel that shows data of selected messages and nodes.
MainWindow	Main window for the program.
NodeChooser	Node chooser panel
RoutingInfoWindow	A window for displaying routing information
SimMenuBar	Menu bar of the simulator GUI

Package gui Description

Contains the classes of Graphical User Interface.

Package gui.playfield

Contains the classes of Graphical User Interface's playfield -view (the graphical presentation of the nodes' locations and other information).

See:

[Description](#)

Class Summary

MapGraphic	PlayfieldGraphic for SimMap visualization
MessageGraphic	Visualization of a message
NodeGraphic	Visualization of a DTN Node
PathGraphic	Visualization of a Path
PlayField	The canvas where node graphics and message visualizations are drawn.
PlayFieldGraphic	Superclass for all graphics to be drawn on the "play field".
ScaleReferenceGraphic	Reference scale bar graphic.

Package gui.playfield Description

Contains the classes of Graphical User Interface's playfield -view (the graphical presentation of the nodes' locations and other information).

Package input

Provides interfaces and classes for reading input data from external sources.

See:

[Description](#)

Interface Summary

EventQueue	Interface for event queues.
ExternalEventsReader	Interface for external event readers.

Class Summary

BinaryEventsReader	Reads External Events from a binary file.
ConnectionEvent	A connection up/down event.
DTN2Events	Delivers bundles from dtnd to ONE.
EventQueueHandler	Handler for managing event queues.
ExternalEvent	Super class for all external events.
ExternalEventsQueue	Queue of external events.
ExternalMovementReader	Reader for ExternalMovement movement model's time-location tuples.
MessageBurstGenerator	Message creation -external events generator.
MessageCreateEvent	External event for creating a message.
MessageDeleteEvent	External event for deleting a message.
MessageEvent	A message related external event
MessageEventGenerator	Message creation -external events generator.
MessageRelayEvent	External event for all the stages of relaying a message between two hosts (start and possible abort or delivery).
OneFromEachMessageGenerator	Message creation -external events generator.
OneToEachMessageGenerator	Message creation -external events generator.
ScheduledUpdatesQueue	Event queue where simulation objects can request an update to happen at the specified simulation time.
StandardEventsReader	External events reader for standard-format events (created e.g by the dtnsim2parser).
WKTMapReader	"Well-known text syntax" map data reader. Note: Understands only <code>LINESTRINGS</code> and <code>MULTILINESTRINGS</code> .
WKTReader	Class for reading "Well-known text syntax" files.

Package input Description

Provides interfaces and classes for reading input data from external sources.

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Package interfaces

Class Summary

ConnectivityGrid	Overlay grid of the world where each interface is put on a cell depending of its location.
ConnectivityOptimizer	A superclass for schemes for optimizing the location of possible contacts with network interfaces of a specific range
InterferenceLimitedInterface	A simple Network Interface that provides a variable bit-rate service, where the bit-rate depends on the number of other transmitting stations within range. The current transmit speed is updated only if there are ongoing transmissions.
SimpleBroadcastInterface	A simple Network Interface that provides a constant bit-rate service, where one transmission can be on at a time.

Package movement

Contains different movement models and related classes for the simulator.

See:

[Description](#)

Interface Summary

SwitchableMovement	Movement models to be used by ExtendedMovementModels should implement this interface
TransportMovement	MovementModels used for transportation should implement this interface

Class Summary

ActivenessHandler	Object of this class tell the movement models when a node belonging to a certain group is active and when not.
BusControlSystem	This class controls busses and passengers that can use the bus.
BusMovement	This class controls the movement of busses.
BusTravellerMovement	This class controls the movement of bus travellers.
CarMovement	The CarMovement class representing the car movement submodel
ClusterMovement	
EveningActivityControlSystem	This class controls the group mobility of the people meeting their friends in the evening
EveningActivityMovement	A Class to model movement when people are out shopping or doing other activities with friends.
EveningTrip	A class to encapsulate information about a shopping trip 1.
ExtendedMovementModel	Classes derived from this can make use of other movement models that implement the SwitchableMovement interface.
ExternalMovement	Movement model that uses external data of node locations.
HomeActivityMovement	A Class to model movement at home.
LinearFormation	A stationary "movement" model where nodes do not move but are in linear formation (i.e., in a line).
MapBasedMovement	Map based movement model which gives out Paths that use the roads of a SimMap.
MapRouteMovement	Map based movement model that uses predetermined paths within the map area.
MovementModel	Superclass for all movement models.
OfficeActivityMovement	This class models movement at an office.
Path	A Path between multiple Coordinates.
RandomWalk	Random Walk movement model
RandomWaypoint	Random waypoint movement model.

ShortestPathMapBasedMovement	Map based movement model that uses Dijkstra's algorithm to find shortest paths between two random map nodes and Points Of Interest
StationaryMovement	A dummy stationary "movement" model where nodes do not move.
WorkingDayMovement	This movement model makes use of several other movement models to simulate movement with daily routines.

Package movement Description

Contains different movement models and related classes for the simulator. All movement models have to be in this package and must extend the [MovementModel](#) class so they can be dynamically loaded to the simulator. The classes to load can be specified through [Settings](#) class' settings source. See MovementModel class and classes extending it for details about the settings. Complex movement models can store their other classes (the ones that don't extend MovementModel class) to sub packages.

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Package movement.map

Sub package for MapBasedMovement movement model's (and its sub classes) helper classes.

See:

[Description](#)

Class Summary

DijkstraPathFinder	Implementation of the Dijkstra's shortest path algorithm.
MapNode	A node in a SimMap.
MapRoute	A route that consists of map nodes.
PointsOfInterest	Handler for points of interest data.
SimMap	A simulation map for node movement.

Package movement.map Description

Sub package for MapBasedMovement movement model's (and its sub classes) helper classes.

Package report

Contains all the report classes.

See:

[Description](#)

Class Summary

AdjacencyGraphvizReport	Generates Graphviz compatible graph from connections.
ConnectivityDtnsim2Report	Link connectivity report generator for DTNSim2 input.
ConnectivityONEReport	Link connectivity report generator for ONE StandardEventsReader input.
ContactsDuringAnICTReport	The number of contacts during an inter-contact time metric is similar to the inter-contact times metric, except that instead of measuring the time until a node meets again, we count the number of other nodes both of the nodes meet separately.
ContactsPerHourReport	This report counts the number of contacts each hour
ContactTimesReport	Reports the node contact time (i.e., how long they were in the range of each other) distribution.
CreatedMessagesReport	Reports information about all created messages.
DeliveredMessagesReport	Report information about all delivered messages.
DistanceDelayReport	Report for how far apart the nodes were when the message was sent and how long time & how many hops it took to deliver it.
DTN2Reporter	The DTN2Reporter class is responsible for delivering bundles from The ONE to dtnd.
EncountersVSUniqueEncountersReport	The total- vs.
EnergyLevelReport	Node energy level report.
EventLogReport	Report that creates same output as the GUI's event log panel but formatted like StandardEventsReader input.
InterContactTimesReport	Reports the inter-contact time (i.e., the time between the end of previous contact and the beginning of a new contact between two hosts) distribution.
MessageDelayReport	Reports delivered messages' delays (one line per delivered message) and cumulative delivery probability sorted by message delays.
MessageDeliveryReport	Report for of amount of messages delivered vs.
MessageGraphvizReport	Creates a graphviz compatible graph of messages that were passed.
MessageLocationReport	Message location report.
MessageReport	Reports delivered messages report: message_id creation_time deliver_time (duplicate)
MessageStatsReport	Report for generating different kind of total statistics about message relaying performance.
	Movement report that generates suitable movement data for ns-2

MovementNs2Report	simulator as described in http://www.isi.edu/nsnam/ns/doc/node174.html .
PingAppReporter	Reporter for the PingApplication.
Report	Abstract superclass for all reports.
TotalContactTimeReport	Report for total amount of contact times among hosts.
TotalEncountersReport	A report of the distribution of how many encounters (contacts) a node has had
UniqueEncountersReport	UniqueEncountersReport class creates a report of the distribution of how many promilles of the other nodes a node has encountered.

Package report Description

Contains all the report classes. Reports can be used to create e.g. statistics and visualizations of the simulation. All report classes must be in this package and must extend the [Report](#) class so they can be dynamically loaded to the simulator. The classes to load can be specified through [Settings](#) class' settings source. See Report class and classes extending it for details about the settings.

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Package routing

Contains all the router classes who decide how to handle the messages.

See:

[Description](#)

Class Summary

ActiveRouter	Superclass of active routers.
DirectDeliveryRouter	Router that will deliver messages only to the final recipient.
EnergyAwareRouter	Energy level-aware variant of Epidemic router.
EpidemicOracleRouter	Epidemic message router with an oracle that tells when a message is delivered and that message is then removed from all nodes that use this routing module.
EpidemicRouter	Epidemic message router with drop-oldest buffer and only single transferring connections at a time.
FirstContactRouter	First contact router which uses only a single copy of the message (or fragments) and forwards it to the first available contact.
MaxPropRouter	Implementation of MaxProp router as described in <i>MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks</i> by John Burgess et al.
MaxPropRouterWithEstimation	Implementation of MaxProp router as described in <i>MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks</i> by John Burgess et al.
MessageRouter	Superclass for message routers.
PassiveRouter	Passive router that doesn't send anything unless commanded.
ProphetRouter	Implementation of PRoPHET router as described in <i>Probabilistic routing in intermittently connected networks</i> by Anders Lindgren et al.
ProphetRouterWithEstimation	Implementation of PRoPHET router as described in <i>Probabilistic routing in intermittently connected networks</i> by Anders Lindgren et al.
RoutingInfo	Class for storing routing related information in a tree form for user interface(s).
SprayAndWaitRouter	Implementation of Spray and wait router as depicted in <i>Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks</i> by Thrasyvoulos Spyropoulos et al.

Package routing Description

Contains all the router classes who decide how to handle the messages. All router classes must be in this package and must extend the [MessageRouter](#) (when creating new routers, extending [ActiveRouter](#) might make sense) class so they can be dynamically loaded to the simulator. The classes to load can be specified through [Settings](#) class' settings source. See MessageRouter class and classes extending it for details about the settings.

Package routing.maxprop

Contains MaxProp routing module specific classes.

See:

[Description](#)

Class Summary

MaxPropDijkstra	Dijkstra's shortest path implementation for MaxProp Router module.
MeetingProbabilitySet	Class for storing and manipulating the meeting probabilities for the MaxProp router module.

Package routing.maxprop Description

Contains MaxProp routing module specific classes.

Package routing.schedule

Class Summary

ScheduleDijkstra	Dijkstra's shortest path implementation for schedule data
ScheduleEntry	
ScheduleOracle	

Package ui

Contains superclass for all user interfaces and a simple user interface(s).

See:

[Description](#)

Class Summary

DTNSimTextUI	Simple text-based user interface.
DTNSimUI	Abstract superclass for user interfaces; contains also some simulation settings.

Package ui Description

Contains superclass for all user interfaces and a simple user interface(s).

movement

Class BusTravellerMovement

```
java.lang.Object
└ movement.MovementModel
  └ movement.MapBasedMovement
    └ movement.BusTravellerMovement
```

All Implemented Interfaces:

[SwitchableMovement](#), [TransportMovement](#)

```
public class BusTravellerMovement
extends MapBasedMovement
implements SwitchableMovement, TransportMovement
```

This class controls the movement of bus travellers. A bus traveller belongs to a bus control system. A bus traveller has a destination and a start location. If the direct path to the destination is longer than the path the node would have to walk if it would take the bus, the node uses the bus. If the destination is not provided, the node will pass a random number of stops determined by Markov chains (defined in settings).

Field Summary

static java.lang.String	PROBABILITIES_STRING
-------------------------	--------------------------------------

static java.lang.String	PROBABILITY TAKE OTHER BUS
-------------------------	--

static int	STATE DECIDED TO ENTER A BUS
------------	--

static int	STATE TRAVELLING ON BUS
------------	---

static int	STATE WAITING FOR BUS
------------	---------------------------------------

static int	STATE WALKING ELSEWHERE
------------	---

Fields inherited from class movement.MapBasedMovement

backAllowed, FILE_S, lastMapNode, MAP_BASE_MOVEMENT_NS, MAP_SELECT_S, maxPathLength, minPathLength, NROF_FILES_S
--

Fields inherited from class movement.MovementModel

comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime, MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
--

Constructor Summary

BusTravellerMovement(BusTravellerMovement proto)
--

Creates a BusTravellerModel from a prototype
--

[**BusTravellerMovement**](#)([Settings](#) settings)
Creates a BusTravellerModel

Method Summary

void	enterBus (Path nextPath) Notifies the node at the bus stop that a bus is there.
protected double	generateWaitTime () Switches state between getPath() calls
int	getID ()
Coord	getInitialLocation () Returns a (random) coordinate that is between two adjacent MapNodes
Coord	getLastLocation () Get the last location the getPath() of this movement model has returned
Coord	getLocation () Get the location where the bus is located when it has moved its path
Path	getPath () Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
int	getState ()
boolean	isReady () Checks if the movement model is finished doing its task and it's time to switch to the next movement model.
MapBasedMovement	replicate () Creates a replicate of the movement model.
static void	reset ()
void	setLocation (Coord lastWaypoint) Tell the movement model what its current location is
void	setNextRoute (Coord nodeLocation, Coord nodeDestination) Sets the next route for the traveller, so that it can decide whether it should take the bus or not.

Methods inherited from class movement.[MapBasedMovement](#)

[getMap](#), [getOkMapNodeTypes](#), [selectRandomOkNode](#)

Methods inherited from class movement.[MovementModel](#)

[generateSpeed](#), [getComBus](#), [getMaxX](#), [getMaxY](#), [isActive](#), [nextPathAvailable](#), [setComBus](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

PROBABILITIES_STRING

```
public static final java.lang.String PROBABILITIES_STRING
```

See Also:

[Constant Field Values](#)

PROBABILITY_TAKE_OTHER_BUS

```
public static final java.lang.String PROBABILITY_TAKE_OTHER_BUS
```

See Also:

[Constant Field Values](#)

STATE_WAITING_FOR_BUS

```
public static final int STATE_WAITING_FOR_BUS
```

See Also:

[Constant Field Values](#)

STATE_DECIDED_TO_ENTER_A_BUS

```
public static final int STATE_DECIDED_TO_ENTER_A_BUS
```

See Also:

[Constant Field Values](#)

STATE_TRAVELLING_ON_BUS

```
public static final int STATE_TRAVELLING_ON_BUS
```

See Also:

[Constant Field Values](#)

STATE_WALKING_ELSEWHERE

```
public static final int STATE_WALKING_ELSEWHERE
```

See Also:

[Constant Field Values](#)

Constructor Detail

BusTravellerMovement

```
public BusTravellerMovement(Settings settings)
```

Creates a BusTravellerModel

Parameters:

settings -

BusTravellerMovement

```
public BusTravellerMovement(BusTravellerMovement proto)
```

Creates a BusTravellerModel from a prototype

Parameters:

proto -

Method Detail

getInitialLocation

```
public Coord getInitialLocation()
```

Description copied from class: [MapBasedMovement](#)

Returns a (random) coordinate that is between two adjacent MapNodes

Overrides:

[getInitialLocation](#) in class [MapBasedMovement](#)

Returns:

The initial coordinates for a node

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Overrides:

[getPath](#) in class [MapBasedMovement](#)

Returns:

A new path or null

generateWaitTime

```
protected double generateWaitTime()
```

Switches state between getPath() calls

Overrides:

[generateWaitTime](#) in class [MovementModel](#)

Returns:

Always 0

replicate

```
public MapBasedMovement replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Overrides:

[replicate](#) in class [MapBasedMovement](#)

Returns:

A new movement model with the same settings as this model

getState

```
public int getState()
```

getLocation

```
public Coord getLocation()
```

Get the location where the bus is located when it has moved its path

Returns:

The end point of the last path returned

enterBus

```
public void enterBus(Path nextPath)
```

Notifies the node at the bus stop that a bus is there. Nodes inside busses are also notified.

Parameters:

nextPath - The next path the bus is going to take

getID

```
public int getID()
```

setNextRoute

```
public void setNextRoute(Coord nodeLocation,
                        Coord nodeDestination)
```

Sets the next route for the traveller, so that it can decide whether it should take the bus or not.

Specified by:

[setNextRoute](#) in interface [TransportMovement](#)

Parameters:

nodeLocation -
nodeDestination -

getLastLocation

```
public Coord getLastLocation()
```

Description copied from interface: [SwitchableMovement](#)

Get the last location the getPath() of this movement model has returned

Specified by:

[getLastLocation](#) in interface [SwitchableMovement](#)

Overrides:

[getLastLocation](#) in class [MapBasedMovement](#)

Returns:

the last location

See Also:

[SwitchableMovement](#)

setLocation

```
public void setLocation(Coord lastWaypoint)
```

Description copied from interface: [SwitchableMovement](#)

Tell the movement model what its current location is

Specified by:

[setLocation](#) in interface [SwitchableMovement](#)

Overrides:

[setLocation](#) in class [MapBasedMovement](#)

See Also:

[SwitchableMovement](#)

isReady

```
public boolean isReady()
```

Description copied from interface: [SwitchableMovement](#)

Checks if the movement model is finished doing its task and it's time to switch to the next movement model. The method should be called between getPath() calls.

Specified by:

[isReady](#) in interface [SwitchableMovement](#)

Overrides:

[isReady](#) in class [MapBasedMovement](#)

Returns:

true if ready

See Also:

[SwitchableMovement](#)

reset

```
public static void reset()
```

movement

Class CarMovement

```
java.lang.Object
└ movement.MovementModel
  └ movement.MapBasedMovement
    └ movement.CarMovement
```

All Implemented Interfaces:

[SwitchableMovement](#), [TransportMovement](#)

```
public class CarMovement
extends MapBasedMovement
implements SwitchableMovement, TransportMovement
```

The CarMovement class representing the car movement submodel

Field Summary

Fields inherited from class movement.MapBasedMovement

```
backAllowed, FILE_S, lastMapNode, MAP_BASE_MOVEMENT_NS, MAP_SELECT_S, maxPathLength,
minPathLength, NROF_FILES_S
```

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

[CarMovement\(CarMovement proto\)](#)

Construct a new CarMovement instance from a prototype

[CarMovement\(Settings settings\)](#)

Car movement constructor

Method Summary

Path	getPath() Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
boolean	isReady() Checks if the movement model is finished doing its task and it's time to switch to the next movement model.
void	setNextRoute(Coord nodeLocation, Coord nodeDestination) Sets the next route to be taken

Methods inherited from class movement.MapBasedMovement

getInitialLocation , getLastLocation , getMap , getOkMapNodeTypes , replicate , selectRandomOkNode , setLocation
--

Methods inherited from class movement.[MovementModel](#)

generateSpeed , generateWaitTime , getComBus , getMaxX , getMaxY , isActive , nextPathAvailable , reset , setComBus , toString
--

Methods inherited from class java.lang.Object

clone , equals , finalize , getClass , hashCode , notify , notifyAll , wait , wait , wait

Methods inherited from interface movement.[SwitchableMovement](#)

getLastLocation , setLocation

Constructor Detail

CarMovement

```
public CarMovement(Settings settings)
```

Car movement constructor

Parameters:

settings -

CarMovement

```
public CarMovement(CarMovement proto)
```

Construct a new CarMovement instance from a prototype

Parameters:

proto -

Method Detail

setNextRoute

```
public void setNextRoute(Coord nodeLocation,  
                         Coord nodeDestination)
```

Sets the next route to be taken

Specified by:

[setNextRoute](#) in interface [TransportMovement](#)

Parameters:

nodeLocation -

nodeDestination -

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Overrides:

[getPath](#) in class [MapBasedMovement](#)

Returns:

A new path or null

isReady

public boolean **isReady()**

Description copied from interface: [SwitchableMovement](#)

Checks if the movement model is finished doing its task and it's time to switch to the next movement model. The method should be called between getPath() calls.

Specified by:

[isReady](#) in interface [SwitchableMovement](#)

Overrides:

[isReady](#) in class [MapBasedMovement](#)

Returns:

true

See Also:

[SwitchableMovement](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Class CBRConnection

```
java.lang.Object
  └ core.Connection
    └ core.CBRConnection
```

```
public class CBRConnection
extends Connection
```

A constant bit-rate connection between two DTN nodes.

Field Summary

Fields inherited from class core.Connection

[bytesTransferred](#), [fromInterface](#), [fromNode](#), [msgFromNode](#), [msgOnFly](#), [toInterface](#), [toNode](#)

Constructor Summary

```
CBRConnection(DTNHost fromNode, NetworkInterface fromInterface, DTNHost toNode,
NetworkInterface toInterface, int connectionSpeed)
```

Creates a new connection between nodes and sets the connection state to "up".

Method Summary

void	abortTransfer() Aborts the transfer of the currently transferred message.
int	getRemainingByteCount() Returns the amount of bytes to be transferred before ongoing transfer is ready or 0 if there's no ongoing transfer or it has finished already
double	getSpeed() returns the current speed of the connection
double	getTransferDoneTime() Gets the transferdonetime
boolean	isMessageTransferred() Returns true if the current message transfer is done.
int	startTransfer(DTNHost from, Message m) Sets a message that this connection is currently transferring.
java.lang.String	toString() Returns a String presentation of the connection.

Methods inherited from class core.Connection

```
clearMsgOnFly, finalizeTransfer, getMessage, getOtherInterface, getOtherNode,
getTotalBytesTransferred, isInitiator, isReadyForTransfer, isUp, setUpState, update
```

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Detail

CBRConnection

```
public CBRConnection(DTNHost fromNode,
                     NetworkInterface fromInterface,
                     DTNHost toNode,
                     NetworkInterface toInterface,
                     int connectionSpeed)
```

Creates a new connection between nodes and sets the connection state to "up".

Parameters:

- fromNode - The node that initiated the connection
- fromInterface - The interface that initiated the connection
- toNode - The node in the other side of the connection
- toInterface - The interface in the other side of the connection
- connectionSpeed - Transfer speed of the connection (Bps) when the connection is initiated

Method Detail

startTransfer

```
public int startTransfer(DTNHost from,
                        Message m)
```

Sets a message that this connection is currently transferring. If message passing is controlled by external events, this method is not needed (but then e.g. [Connection.finalizeTransfer\(\)](#) and [isMessageTransferred\(\)](#) will not work either). Only a one message at a time can be transferred using one connection.

Specified by:

[startTransfer](#) in class [Connection](#)

Parameters:

- from - The host sending the message
- m - The message

Returns:

The value returned by [MessageRouter.receiveMessage\(Message, DTNHost\)](#)

abortTransfer

```
public void abortTransfer()
```

Aborts the transfer of the currently transferred message.

Overrides:

[abortTransfer](#) in class [Connection](#)

getTransferDoneTime

```
public double getTransferDoneTime()
```

Gets the transferdonetime

isMessageTransferred

```
public boolean isMessageTransferred()
```

Returns true if the current message transfer is done.

Specified by:

[isMessageTransferred](#) in class [Connection](#)

Returns:

True if the transfer is done, false if not

getSpeed

```
public double getSpeed()
```

returns the current speed of the connection

Specified by:

[getSpeed](#) in class [Connection](#)

getRemainingByteCount

```
public int getRemainingByteCount()
```

Returns the amount of bytes to be transferred before ongoing transfer is ready or 0 if there's no ongoing transfer or it has finished already

Specified by:

[getRemainingByteCount](#) in class [Connection](#)

Returns:

the amount of bytes to be transferred

toString

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

Returns a String presentation of the connection.

Overrides:

[toString](#) in class [Connection](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement

Class ClusterMovement

```
java.lang.Object
  └── movement.MovementModel
    └── movement.RandomWaypoint
      └── movement.ClusterMovement
```

```
public class ClusterMovement
extends RandomWaypoint
```

Field Summary

static java.lang.String	CLUSTER_CENTER Center point of the cluster
static java.lang.String	CLUSTER_RANGE Range of the cluster

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

[ClusterMovement](#)(Settings s)

Method Summary

int	getMaxX() Returns the largest X coordinate value this model uses
int	getMaxY() Returns the largest Y coordinate value this model uses
protected Coord	randomCoord()
ClusterMovement	replicate() Creates a replicate of the movement model.

Methods inherited from class movement.RandomWaypoint

[getInitialLocation](#), [getPath](#)

Methods inherited from class movement.MovementModel

```
generateSpeed, generateWaitTime, getComBus, isActive, nextPathAvailable, reset, setComBus,
toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Field Detail**CLUSTER_RANGE**

```
public static final java.lang.String CLUSTER_RANGE
```

Range of the cluster

See Also:

[Constant Field Values](#)

CLUSTER_CENTER

```
public static final java.lang.String CLUSTER_CENTER
```

Center point of the cluster

See Also:

[Constant Field Values](#)

Constructor Detail**ClusterMovement**

```
public ClusterMovement(Settings s)
```

Method Detail**randomCoord**

```
protected Coord randomCoord()
```

Overrides:

[randomCoord](#) in class [RandomWaypoint](#)

getMaxX

```
public int getMaxX()
```

Description copied from class: [MovementModel](#)

Returns the largest X coordinate value this model uses

Overrides:

[getMaxX](#) in class [MovementModel](#)

Returns:

Maximum of X coordinate values

getMaxY

```
public int getMaxY()
```

Description copied from class: [MovementModel](#)

Returns the largest Y coordinate value this model uses

Overrides:

[getMaxY](#) in class [MovementModel](#)

Returns:

Maximum of Y coordinate values

replicate

```
public ClusterMovement replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Overrides:

[replicate](#) in class [RandomWaypoint](#)

Returns:

A new movement model with the same settings as this model

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Class Connection

```
java.lang.Object
└ core.Connection
```

Direct Known Subclasses:[CBRConnection](#), [VBRCConnection](#)

```
public abstract class Connection
extends java.lang.Object
```

A connection between two DTN nodes.

Field Summary

protected int	bytesTransferred how many bytes this connection has transferred
protected NetworkInterface	fromInterface
protected DTNHost	fromNode
protected DTNHost	msgFromNode
protected Message	msgOnFly
protected NetworkInterface	toInterface
protected DTNHost	toNode

Constructor Summary

```
Connection(DTNHost fromNode, NetworkInterface fromInterface, DTNHost toNode,
NetworkInterface toInterface)
```

Creates a new connection between nodes and sets the connection state to "up".

Method Summary

void	abortTransfer() Aborts the transfer of the currently transferred message.
protected void	clearMsgOnFly() Clears the message that is currently being transferred.
void	finalizeTransfer() Finalizes the transfer of the currently transferred message.

	Message getMessage() Gets the message that this connection is currently transferring.
NetworkInterface	getOtherInterface(NetworkInterface i) Returns the interface in the other end of the connection
DTNHost	getOtherNode(DTNHost node) Returns the node in the other end of the connection
abstract int	getRemainingByteCount() Returns the amount of bytes to be transferred before ongoing transfer is ready or 0 if there's no ongoing transfer or it has finished already
abstract double	getSpeed() Gets the current connection speed
int	getTotalBytesTransferred() Returns the total amount of bytes this connection has transferred so far (including all transfers).
boolean	isInitiator(DTNHost node) Returns true if the given node is the initiator of the connection, false otherwise
abstract boolean	isMessageTransferred() Returns true if the current message transfer is done
boolean	isReadyForTransfer() Returns true if the connection is ready to transfer a message (connection is up and there is no message being transferred).
boolean	isUp() Returns true if the connection is up
void	setUpState(boolean state) Sets the state of the connection.
abstract int	startTransfer(DTNHost from, Message m) Sets a message that this connection is currently transferring.
java.lang.String	toString() Returns a String presentation of the connection.
void	update() Calculate the current transmission speed from the information given by the interfaces, and calculate the missing data amount.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail**toNode**protected [DTNHost](#) toNode**toInterface**protected [NetworkInterface](#) toInterface

fromNode

```
protected DTNHost fromNode
```

fromInterface

```
protected NetworkInterface fromInterface
```

msgFromNode

```
protected DTNHost msgFromNode
```

msgOnFly

```
protected Message msgOnFly
```

bytesTransferred

```
protected int bytesTransferred
```

how many bytes this connection has transferred

Constructor Detail**Connection**

```
public Connection(DTNHost fromNode,
                  NetworkInterface fromInterface,
                  DTNHost toNode,
                  NetworkInterface toInterface)
```

Creates a new connection between nodes and sets the connection state to "up".

Parameters:

- fromNode - The node that initiated the connection
- fromInterface - The interface that initiated the connection
- toNode - The node in the other side of the connection
- toInterface - The interface in the other side of the connection

Method Detail**isUp**

```
public boolean isUp()
```

Returns true if the connection is up

Returns:

- state of the connection

isInitiator

```
public boolean isInitiator(DTNHost node)
```

Returns true if the given node is the initiator of the connection, false otherwise

Parameters:

node - The node to check

Returns:

true if the given node is the initiator of the connection

setUpState

```
public void setUpState(boolean state)
```

Sets the state of the connection.

Parameters:

state - True if the connection is up, false if not

startTransfer

```
public abstract int startTransfer(DTNHost from,  
Message m)
```

Sets a message that this connection is currently transferring. If message passing is controlled by external events, this method is not needed (but then e.g. [finalizeTransfer\(\)](#) and [isMessageTransferred\(\)](#) will not work either). Only a one message at a time can be transferred using one connection.

Parameters:

m - The message

Returns:

The value returned by [MessageRouter.receiveMessage\(Message, DTNHost\)](#)

update

```
public void update( )
```

Calculate the current transmission speed from the information given by the interfaces, and calculate the missing data amount.

abortTransfer

```
public void abortTransfer( )
```

Aborts the transfer of the currently transferred message.

getRemainingByteCount

```
public abstract int getRemainingByteCount( )
```

Returns the amount of bytes to be transferred before ongoing transfer is ready or 0 if there's no ongoing transfer or it has finished already

Returns:

the amount of bytes to be transferred

clearMsgOnFly

```
protected void clearMsgOnFly()
```

Clears the message that is currently being transferred. Calls to [getMessage\(\)](#) will return null after this.

finalizeTransfer

```
public void finalizeTransfer()
```

Finalizes the transfer of the currently transferred message. The message that was being transferred can **not** be retrieved from this connections after calling this method (using [getMessage\(\)](#)).

isMessageTransferred

```
public abstract boolean isMessageTransferred()
```

Returns true if the current message transfer is done

Returns:

True if the transfer is done, false if not

isReadyForTransfer

```
public boolean isReadyForTransfer()
```

Returns true if the connection is ready to transfer a message (connection is up and there is no message being transferred).

Returns:

true if the connection is ready to transfer a message

getMessage

```
public Message getMessage()
```

Gets the message that this connection is currently transferring.

Returns:

The message or null if no message is being transferred

getSpeed

```
public abstract double getSpeed()
```

Gets the current connection speed

getTotalBytesTransferred

```
public int getTotalBytesTransferred()
```

Returns the total amount of bytes this connection has transferred so far (including all transfers).

getOtherNode

```
public DTNHost getOtherNode(DTNHost node)
```

Returns the node in the other end of the connection

Parameters:

node - The node in this end of the connection

Returns:

The requested node

getOtherInterface

```
public NetworkInterface getOtherInterface(NetworkInterface i)
```

Returns the interface in the other end of the connection

Parameters:

i - The interface in this end of the connection

Returns:

The requested interface

toString

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

Returns a String presentation of the connection.

Overrides:

`toString` in class `java.lang.Object`

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class ConnectionEvent

```
java.lang.Object
  └── input.ExternalEvent
      └── input.ConnectionEvent
```

All Implemented Interfaces:java.io.Serializable, java.lang.Comparable<[ExternalEvent](#)>

```
public class ConnectionEvent
extends ExternalEvent
```

A connection up/down event.

See Also:[Serialized Form](#)

Field Summary

protected int	fromAddr address of the node the (dis)connection is from
protected java.lang.String	interfaceId What is the interface number for this event
protected boolean	isUp Is this a "connection up" event
protected int	toAddr address of the node the (dis)connection is to

Fields inherited from class input.ExternalEvent

[time](#)

Constructor Summary

```
ConnectionEvent(int from, int to, java.lang.String interf, boolean up, double time)
Creates a new connection event
```

Method Summary

void	processEvent (World world) Processes the external event.
java.lang.String	toString () Returns a String representation of the event

Methods inherited from class input.ExternalEvent

[compareTo](#), [getTime](#)

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Field Detail**fromAddr**

```
protected int fromAddr
```

address of the node the (dis)connection is from

toAddr

```
protected int toAddr
```

address of the node the (dis)connection is to

isUp

```
protected boolean isUp
```

Is this a "connection up" event

interfaceId

```
protected java.lang.String interfaceId
```

What is the interface number for this event

Constructor Detail**ConnectionEvent**

```
public ConnectionEvent(int from,
                      int to,
                      java.lang.String interf,
                      boolean up,
                      double time)
```

Creates a new connection event

Parameters:

- from - End point of connection
- to - Another end of connection
- interf - The number of interface for the connection
- up - If true, this was a "connection up" event, if false, this was a "connection down" event
- time - Time when the Connection event occurs

Method Detail

processEvent

```
public void processEvent(World world)
```

Description copied from class: [ExternalEvent](#)

Processes the external event.

Overrides:

[processEvent](#) in class [ExternalEvent](#)

Parameters:

world - World where the actors of the event are

toString

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

Description copied from class: [ExternalEvent](#)

Returns a String representation of the event

Overrides:

[toString](#) in class [ExternalEvent](#)

Returns:

a String representation of the event

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Interface ConnectionListener

All Known Implementing Classes:

[AdjacencyGraphvizReport](#), [ConnectivityDtnsim2Report](#), [ConnectivityONEReport](#),
[ContactsDuringAnICTReport](#), [ContactsPerHourReport](#), [ContactTimesReport](#),
[EncountersVSUniqueEncountersReport](#), [EventLogPanel](#), [EventLogReport](#), [InterContactTimesReport](#),
[TotalContactTimeReport](#), [TotalEncountersReport](#), [UniqueEncountersReport](#)

```
public interface ConnectionListener
```

Interface for classes that want to be informed about connections between hosts.

Method Summary

void	hostsConnected (DTNHost host1, DTNHost host2)
	Method is called when two hosts are connected.
void	hostsDisconnected (DTNHost host1, DTNHost host2)
	Method is called when connection between hosts is disconnected.

Method Detail

hostsConnected

```
void hostsConnected(DTNHost host1,
                    DTNHost host2)
```

Method is called when two hosts are connected.

Parameters:

host1 - Host that initiated the connection
host2 - Host that was connected to

hostsDisconnected

```
void hostsDisconnected(DTNHost host1,
                      DTNHost host2)
```

Method is called when connection between hosts is disconnected.

Parameters:

host1 - Host that initiated the disconnection
host2 - Host at the other end of the connection

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class ConnectivityDtnsim2Report

```
java.lang.Object
└─ report.Report
    └─ report.ConnectivityDtnsim2Report
```

All Implemented Interfaces:[ConnectionListener](#)

```
public class ConnectivityDtnsim2Report
extends Report
implements ConnectionListener
```

Link connectivity report generator for DTNSim2 input. Connections that start during the warm up period are ignored.

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[ConnectivityDtnsim2Report\(\)](#)

Constructor.

Method Summary

void	hostsConnected(DTNHost h1, DTNHost h2) Method is called when two hosts are connected.
------	--

void	hostsDisconnected(DTNHost h1, DTNHost h2) Method is called when connection between hosts is disconnected.
------	--

Methods inherited from class report.Report

```
addWarmupID, done, format, getAverage, getIntAverage, getIntMedian, getMedian,
getScenarioName, getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent,
removeWarmupID, setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

ConnectivityDtnsim2Report

```
public ConnectivityDtnsim2Report()
```

Constructor.

Method Detail

hostsConnected

```
public void hostsConnected(DTNHost h1,  
                           DTNHost h2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

h1 - Host that initiated the connection
h2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost h1,  
                             DTNHost h2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

h1 - Host that initiated the disconnection
h2 - Host at the other end of the connection

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

interfaces

Class ConnectivityGrid

```
java.lang.Object
  ↘ interfaces.ConnectivityOptimizer
    ↘ interfaces.ConnectivityGrid
```

```
public class ConnectivityGrid
extends ConnectivityOptimizer
```

Overlay grid of the world where each interface is put on a cell depending of its location. This is used in cell-based optimization of connecting the interfaces.

The idea in short:

Instead of checking for every interface if some of the other interfaces are close enough (this approach obviously doesn't scale) we check only interfaces that are "close enough" to be possibly connected. Being close enough is determined by keeping track of the approximate location of the interfaces by storing them in overlay grid's cells and updating the cell information every time the interfaces move. If two interfaces are in the same cell or in neighboring cells, they have a chance of being close enough for connection. Then only that subset of interfaces is checked for possible connectivity.

Note: this class does NOT support negative coordinates. Also, it makes sense to normalize the coordinates to start from zero to conserve memory.

Nested Class Summary

class	ConnectivityGrid.GridCell
-------	---

A single cell in the cell grid.

Method Summary

void	addInterface(NetworkInterface ni) Adds a network interface to the overlay grid
void	addInterfaces(java.util.Collection<NetworkInterface> interfaces) Adds interfaces to overlay grid
static ConnectivityGrid	ConnectivityGridFactory(int key, double cellSize) Returns a connectivity grid object based on a hash value
java.util.Collection<NetworkInterface>	getAllInterfaces() Returns all interfaces that use the same technology and channel
java.util.Collection<NetworkInterface>	getNearInterfaces(NetworkInterface netinterf) Returns all interfaces using the same technology and channel that are in neighboring cells
void	removeInterface(NetworkInterface ni) Removes a network interface from the overlay grid
static void	reset()
java.lang.String	toString() Returns a string representation of the ConnectivityCells object
void	updateLocation(NetworkInterface ni) Checks and updates (if necessary) interface's position in the grid

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Method Detail

reset

```
public static void reset()
```

ConnectivityGridFactory

```
public static ConnectivityGrid ConnectivityGridFactory(int key,  
double cellSize)
```

Returns a connectivity grid object based on a hash value

Parameters:

key - A hash value that separates different interfaces from each other

cellSize - Cell's edge's length (must be larger than the largest radio coverage's diameter)

Returns:

The connectivity grid object for a specific interface

addInterface

```
public void addInterface(NetworkInterface ni)
```

Adds a network interface to the overlay grid

Specified by:

[addInterface](#) in class [ConnectivityOptimizer](#)

Parameters:

ni - The new network interface

removeInterface

```
public void removeInterface(NetworkInterface ni)
```

Removes a network interface from the overlay grid

Parameters:

ni - The interface to be removed

addInterfaces

```
public void addInterfaces(java.util.Collection<NetworkInterface> interfaces)
```

Adds interfaces to overlay grid

Specified by:

[addInterfaces](#) in class [ConnectivityOptimizer](#)

Parameters:

interfaces - Collection of interfaces to add

updateLocation

```
public void updateLocation(NetworkInterface ni)
```

Checks and updates (if necessary) interface's position in the grid

Specified by:

[updateLocation](#) in class [ConnectivityOptimizer](#)

Parameters:

ni - The interface to update

getAllInterfaces

```
public java.util.Collection<NetworkInterface> getAllInterfaces()
```

Returns all interfaces that use the same technology and channel

Specified by:

[getAllInterfaces](#) in class [ConnectivityOptimizer](#)

getNearInterfaces

```
public java.util.Collection<NetworkInterface> getNearInterfaces(NetworkInterface netinterf)
```

Returns all interfaces using the same technology and channel that are in neighboring cells

Specified by:

[getNearInterfaces](#) in class [ConnectivityOptimizer](#)

Parameters:

netinterf - network interface that needs to be connected

Returns:

A collection of network interfaces within proximity

toString

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

Returns a string representation of the ConnectivityCells object

Overrides:

[toString](#) in class [java.lang.Object](#)

Returns:

a string representation of the ConnectivityCells object

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class ConnectivityONEReport

```
java.lang.Object
└ report.Report
    └ report.ConnectivityONEReport
```

All Implemented Interfaces:

[ConnectionListener](#)

```
public class ConnectivityONEReport
extends Report
implements ConnectionListener
```

Link connectivity report generator for ONE StandardEventsReader input. Connections that start during the warm up period are ignored.

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[ConnectivityONEReport\(\)](#)

Constructor.

Method Summary

void	hostsConnected(DTNHost h1, DTNHost h2) Method is called when two hosts are connected.
------	--

void	hostsDisconnected(DTNHost h1, DTNHost h2) Method is called when connection between hosts is disconnected.
------	--

Methods inherited from class report.Report

```
addWarmupID, done, format, getAverage, getIntAverage, getIntMedian, getMedian,
getScenarioName, getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent,
removeWarmupID, setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

ConnectivityONEReport

```
public ConnectivityONEReport()
```

Constructor.

Method Detail

hostsConnected

```
public void hostsConnected(DTNHost h1,  
                           DTNHost h2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

h1 - Host that initiated the connection

h2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost h1,  
                             DTNHost h2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

h1 - Host that initiated the disconnection

h2 - Host at the other end of the connection

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

interfaces

Class ConnectivityOptimizer

```
java.lang.Object
└─ interfaces.ConnectivityOptimizer
```

Direct Known Subclasses:

[ConnectivityGrid](#)

```
public abstract class ConnectivityOptimizer
extends java.lang.Object
```

A superclass for schemes for optimizing the location of possible contacts with network interfaces of a specific range

Constructor Summary

[ConnectivityOptimizer\(\)](#)

Method Summary

<pre>abstract void</pre>	addInterface(NetworkInterface ni) Adds a network interface to the optimizer (unless it is already present)
<pre>abstract void</pre>	addInterfaces(java.util.Collection<NetworkInterface> interfaces) Adds a collection of network interfaces to the optimizer (except of those already added)
<pre>abstract java.util.Collection<NetworkInterface></pre>	getAllInterfaces() Finds all other interfaces that are registered to the ConnectivityOptimizer
<pre>abstract java.util.Collection<NetworkInterface></pre>	getNearInterfaces(NetworkInterface ni) Finds all network interfaces that might be located so that they can be connected with the network interface
<pre>abstract void</pre>	updateLocation(NetworkInterface ni) Updates a network interface's location

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

ConnectivityOptimizer

```
public ConnectivityOptimizer()
```

Method Detail

addInterface

```
public abstract void addInterface(NetworkInterface ni)
```

Adds a network interface to the optimizer (unless it is already present)

addInterfaces

```
public abstract void addInterfaces(java.util.Collection<NetworkInterface> interfaces)
```

Adds a collection of network interfaces to the optimizer (except of those already added)

updateLocation

```
public abstract void updateLocation(NetworkInterface ni)
```

Updates a network interface's location

getNearInterfaces

```
public abstract java.util.Collection<NetworkInterface> getNearInterfaces(NetworkInterface ni)
```

Finds all network interfaces that might be located so that they can be connected with the network interface

Parameters:

ni - network interface that needs to be connected

Returns:

A collection of network interfaces within proximity

getAllInterfaces

```
public abstract java.util.Collection<NetworkInterface> getAllInterfaces()
```

Finds all other interfaces that are registered to the ConnectivityOptimizer

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class ContactsDuringAnICTReport

```
java.lang.Object
└ report.Report
    └ report.ContactsDuringAnICTReport
```

All Implemented Interfaces:

[ConnectionListener](#), [UpdateListener](#)

```
public class ContactsDuringAnICTReport
extends Report
implements ConnectionListener, UpdateListener
```

The number of contacts during an inter-contact time metric is similar to the inter-contact times metric, except that instead of measuring the time until a node meets again, we count the number of other nodes both of the nodes meet separately. In contrast to the inter-contact times, the number of contacts during an inter-contact is not symmetric, i.e. during an inter-contact both nodes wait the exact same time but will meet a different number of nodes.

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[ContactsDuringAnICTReport\(\)](#)

Method Summary

<code>void</code>	done()	Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
<code>void</code>	hostsConnected(DTNHost host1, DTNHost host2)	Method is called when two hosts are connected.
<code>void</code>	hostsDisconnected(DTNHost host1, DTNHost host2)	Method is called when connection between hosts is disconnected.
<code>protected void</code>	init()	Initializes the report output.
<code>void</code>	updated(java.util.List<DTNHost> hosts)	Method is called on every update cycle.

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
```

```
getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,  
setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

ContactsDuringAnICTReport

```
public ContactsDuringAnICTReport()
```

Method Detail

init

```
protected void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

hostsConnected

```
public void hostsConnected(DTNHost host1,  
                           DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the connection

host2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost host1,  
                            DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

updated

```
public void updated(java.util.List<DTNHost> hosts)
```

Description copied from interface: [UpdateListener](#)

Method is called on every update cycle.

Specified by:

[updated](#) in interface [UpdateListener](#)

Parameters:

hosts - A list of all hosts in the world

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class ContactsPerHourReport

```
java.lang.Object
└ report.Report
  └ report.ContactsPerHourReport
```

All Implemented Interfaces:[ConnectionListener](#)

```
public class ContactsPerHourReport
extends Report
implements ConnectionListener
```

This report counts the number of contacts each hour

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[ContactsPerHourReport\(\)](#)

Method Summary

void	done()	Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
void	hostsConnected(DTNHost host1, DTNHost host2)	Method is called when two hosts are connected.
void	hostsDisconnected(DTNHost host1, DTNHost host2)	Method is called when connection between hosts is disconnected.
void	init()	Initializes the report output.

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,
setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

Constructor Detail

ContactsPerHourReport

```
public ContactsPerHourReport()
```

Method Detail

init

```
public void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

hostsConnected

```
public void hostsConnected(DTNHost host1,
                           DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the connection

host2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost host1,
                             DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class ContactTimesReport

```
java.lang.Object
  └ report.Report
    └ report.ContactTimesReport
```

All Implemented Interfaces:[ConnectionListener](#)**Direct Known Subclasses:**[InterContactTimesReport](#), [TotalContactTimeReport](#)

```
public class ContactTimesReport
extends Report
implements ConnectionListener
```

Reports the node contact time (i.e., how long they were in the range of each other) distribution. Report file contains the count of connections that lasted for certain amount of time. Syntax:

time nrofContacts

Nested Class Summary

protected class	ContactTimesReport.ConnectionInfo	Objects of this class store time information about contacts.
-----------------	---	--

Field Summary

java.util.HashMap< ContactTimesReport.ConnectionInfo , ContactTimesReport.ConnectionInfo >	protected	connections
	protected double	granularity How many seconds are grouped in one group
	static java.lang.String	GRANULARITY Granularity -setting id ("granularity").

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[ContactTimesReport\(\)](#)

Constructor.

Method Summary

protected void	addConnection(DTNHost host1, DTNHost host2)
void	done() Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
void	hostsConnected(DTNHost host1, DTNHost host2) Method is called when two hosts are connected.
void	hostsDisconnected(DTNHost host1, DTNHost host2) Method is called when connection between hosts is disconnected.
protected void	increaseTimeCount(double time) Increases the amount of times a certain time value has been seen.
protected void	init() Initializes the report output.
protected ContactTimesReport.ConnectionInfo	removeConnection(DTNHost host1, DTNHost host2)

Methods inherited from class [report.Report](#)

[addWarmupID](#), [format](#), [getAverage](#), [getIntAverage](#), [getIntMedian](#), [getMedian](#), [getScenarioName](#), [getSettings](#), [getSimTime](#), [getVariance](#), [isWarmup](#), [isWarmupID](#), [newEvent](#), [removeWarmupID](#), [setPrefix](#), [write](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Field Detail

connections

```
protected java.util.HashMap<ContactTimesReport.ConnectionInfo,ContactTimesReport.ConnectionInfo>
connections
```

GRANULARITY

```
public static final java.lang.String GRANULARITY
```

Granularity -setting id ("granularity"). Defines how many simulated seconds are grouped in one reported interval.

See Also:

[Constant Field Values](#)

granularity

```
protected double granularity
```

How many seconds are grouped in one group

Constructor Detail

ContactTimesReport

```
public ContactTimesReport()
```

Constructor.

Method Detail

init

```
protected void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

hostsConnected

```
public void hostsConnected(DTNHost host1,
                           DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the connection

host2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost host1,
                             DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

addConnection

```
protected void addConnection(DTNHost host1,
```

[DTNHost](#) host2)

removeConnection

```
protected ContactTimesReport.ConnectionInfo removeConnection(DTNHost host1,  
DTNHost host2)
```

increaseTimeCount

```
protected void increaseTimeCount(double time)
```

Increases the amount of times a certain time value has been seen.

Parameters:

time - The time value that was seen

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Class Coord

```
java.lang.Object
└ core.Coord
```

All Implemented Interfaces:

`java.lang.Cloneable, java.lang.Comparable<Coord>`

```
public class Coord
extends java.lang.Object
implements java.lang.Cloneable, java.lang.Comparable<Coord>
```

Class to hold 2D coordinates and perform simple arithmetics and transformations

Constructor Summary

`Coord(double x, double y)`

Constructor.

Method Summary

<code>Coord</code>	<code>clone()</code> Returns a clone of this coordinate
<code>int</code>	<code>compareTo(Coord other)</code> Compares this coordinate to other coordinate.
<code>double</code>	<code>distance(Coord other)</code> Returns the distance to another coordinate
<code>boolean</code>	<code>equals(Coord c)</code> Checks if this coordinate's location is equal to other coordinate's
<code>boolean</code>	<code>equals(java.lang.Object o)</code>
<code>double</code>	<code>getX()</code> Returns the x coordinate
<code>double</code>	<code>getY()</code> Returns the y coordinate
<code>int</code>	<code>hashCode()</code> Returns a hash code for this coordinate (actually a hash of the String made of the coordinates)
<code>void</code>	<code>setLocation(Coord c)</code> Sets this coordinate's location to be equal to other coordinates location
<code>void</code>	<code>setLocation(double x, double y)</code> Sets the location of this coordinate object
<code>java.lang.String</code>	<code>toString()</code> Returns a text representation of the coordinate (rounded to 2 decimals)

```
void translate(double dx, double dy)
    Moves the point by dx and dy
```

Methods inherited from class java.lang.Object

`finalize, getClass, notify, notifyAll, wait, wait, wait`

Constructor Detail

Coord

```
public Coord(double x,
             double y)
```

Constructor.

Parameters:

- x - Initial X-coordinate
- y - Initial Y-coordinate

Method Detail

setLocation

```
public void setLocation(double x,
                      double y)
```

Sets the location of this coordinate object

Parameters:

- x - The x coordinate to set
- y - The y coordinate to set

setLocation

```
public void setLocation(Coord c)
```

Sets this coordinate's location to be equal to other coordinates location

Parameters:

- c - The other coordinate

translate

```
public void translate(double dx,
                     double dy)
```

Moves the point by dx and dy

Parameters:

- dx - How much to move the point in X-direction
- dy - How much to move the point in Y-direction

distance

```
public double distance(Coord other)
```

Returns the distance to another coordinate

Parameters:

other - The other coordinate

Returns:

The distance between this and another coordinate

getX

```
public double getX()
```

Returns the x coordinate

Returns:

x coordinate

getY

```
public double getY()
```

Returns the y coordinate

Returns:

y coordinate

toString

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

Returns a text representation of the coordinate (rounded to 2 decimals)

Overrides:

`toString` in class `java.lang.Object`

Returns:

a text representation of the coordinate

clone

```
public Coord clone()
```

Returns a clone of this coordinate

Overrides:

`clone` in class `java.lang.Object`

equals

```
public boolean equals(Coord c)
```

Checks if this coordinate's location is equal to other coordinate's

Parameters:

c - The other coordinate

Returns:

True if locations are the same

equals

```
public boolean equals(java.lang.Object o)
```

Overrides:

equals in class java.lang.Object

hashCode

```
public int hashCode()
```

Returns a hash code for this coordinate (actually a hash of the String made of the coordinates)

Overrides:

hashCode in class java.lang.Object

compareTo

```
public int compareTo(Coord other)
```

Compares this coordinate to other coordinate. Coordinate whose y value is smaller comes first and if y values are equal, the one with smaller x value comes first.

Specified by:

compareTo in interface java.lang.Comparable<Coord>

Returns:

-1, 0 or 1 if this node is before, in the same place or after the other coordinate

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class CreatedMessagesReport

```
java.lang.Object
└ report.Report
    └ report.CreatedMessagesReport
```

All Implemented Interfaces:

[MessageListener](#)

```
public class CreatedMessagesReport
extends Report
implements MessageListener
```

Reports information about all created messages. Messages created during the warm up period are ignored. For output syntax, see [HEADER](#).

Field Summary

static java.lang.String	HEADER
-------------------------	------------------------

Fields inherited from class report.Report

DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING, PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime

Constructor Summary

[CreatedMessagesReport\(\)](#)

Constructor.

Method Summary

void	done() Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
void	init() Initializes the report output.
void	messageDeleted(Message m, DTNHost where, boolean dropped) Method is called when a message is deleted
void	messageTransferAborted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer was aborted before it finished
void	messageTransferred(Message m, DTNHost f, DTNHost t, boolean b) Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer is started
void	

[**newMessage \(Message m\)**](#)

Method is called when a new message is created

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,  

getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,  

setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail**HEADER**

```
public static java.lang.String HEADER
```

Constructor Detail**CreatedMessagesReport**

```
public CreatedMessagesReport\(\)
```

Constructor.

Method Detail**init**

```
public void init\(\)
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

newMessage

```
public void newMessage \(Message m\)
```

Description copied from interface: [MessageListener](#)

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

m - Message that was created

messageTransferred

```
public void messageTransferred(Message m,
                             DTNHost f,
                             DTNHost t,
                             boolean b)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

- m - The message that was transferred
- f - Node where the message was transferred from
- t - Node where the message was transferred to
- b - Was the target node final destination of the message and received this message for the first time.

messageDeleted

```
public void messageDeleted(Message m,
                           DTNHost where,
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

- m - The message that was deleted
- where - The host where the message was deleted
- dropped - True if the message was dropped, false if removed

messageTransferAborted

```
public void messageTransferAborted(Message m,
                                   DTNHost from,
                                   DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

- m - The message that was being transferred
- from - Node where the message was being transferred from
- to - Node where the message was being transferred to

messageTransferStarted

```
public void messageTransferStarted(Message m,
                                   DTNHost from,
                                   DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

m - The message that is going to be transferred
from - Node where the message is transferred from
to - Node where the message is transferred to

done

public void **done**()

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Class Debug

```
java.lang.Object
└ core.Debug
```

```
public class Debug
extends java.lang.Object
```

Debugging info printer with time stamping. This class is not to be actively used but convenient for temporary debugging.

Constructor Summary

[Debug\(\)](#)

Method Summary

<code>static void</code>	<code>doneTiming()</code> End timing an action.
<code>static void</code>	<code>p(java.lang.String txt)</code> Prints text to output with level 0
<code>static void</code>	<code>p(java.lang.String txt, int level)</code> Prints text to output given with level
<code>static void</code>	<code>p(java.lang.String txt, int level, boolean timestamp)</code> Print text to debug output.
<code>static void</code>	<code>pt(java.lang.String txt)</code> Debug print with a timestamp and 0 level
<code>static void</code>	<code>pt(java.lang.String txt, int level)</code> Debug print with a timestamp
<code>void</code>	<code>setDebugLevel(int level)</code> Sets the current debug level (smaller level -> more messages)
<code>void</code>	<code>setPrintStream(java.io.PrintStream outStrm)</code> Sets print stream of debug output.
<code>static void</code>	<code>startTiming(java.lang.String cause)</code> Start timing an action.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

Debug

```
public Debug()
```

Method Detail

setDebugLevel

```
public void setDebugLevel(int level)
```

Sets the current debug level (smaller level -> more messages)

Parameters:

level - The level to set

setPrintStream

```
public void setPrintStream(java.io.PrintStream outStrm)
```

Sets print stream of debug output.

Parameters:

outStrm - The stream

p

```
public static void p(java.lang.String txt)
```

Prints text to output with level 0

Parameters:

txt - text to print

p

```
public static void p(java.lang.String txt,  
                     int level)
```

Prints text to output given with level

Parameters:

level - The debug level

txt - text to print

pt

```
public static void pt(java.lang.String txt,  
                     int level)
```

Debug print with a timestamp

Parameters:

txt - Text to print

level - Debug level

pt

```
public static void pt(java.lang.String txt)
```

Debug print with a timestamp and 0 level

Parameters:

txt - Text to print

p

```
public static void p(java.lang.String txt,
                     int level,
                     boolean timestamp)
```

Print text to debug output.

Parameters:

txt - The text to print

level - The debug level (only messages with level >= debugLevel are printed)

timestamp - If true, text is (sim)timestamped

startTiming

```
public static void startTiming(java.lang.String cause)
```

Start timing an action.

See Also:

[doneTiming\(\)](#)

doneTiming

```
public static void doneTiming()
```

End timing an action. Information how long the action took is printed to debug stream.

See Also:

[startTiming\(String\)](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class DeliveredMessagesReport

```
java.lang.Object
└ report.Report
  └ report.DeliveredMessagesReport
```

All Implemented Interfaces:[MessageListener](#)

```
public class DeliveredMessagesReport
extends Report
implements MessageListener
```

Report information about all delivered messages. Messages created during the warm up period are ignored. For output syntax, see [HEADER](#).

Field Summary

static java.lang.String	HEADER
-------------------------	------------------------

Fields inherited from class report.Report

DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING, PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime

Constructor Summary

[DeliveredMessagesReport\(\)](#)

Constructor.

Method Summary

void	done() Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
void	init() Initializes the report output.
void	messageDeleted(Message m, DTNHost where, boolean dropped) Method is called when a message is deleted
void	messageTransferAborted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer was aborted before it finished
void	messageTransferred(Message m, DTNHost from, DTNHost to, boolean firstDelivery) Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer is started
void	

[newMessage \(Message m\)](#)

Method is called when a new message is created

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,
setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail**HEADER**

```
public static java.lang.String HEADER
```

Constructor Detail**DeliveredMessagesReport**

```
public DeliveredMessagesReport()
```

Constructor.

Method Detail**init**

```
public void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

messageTransferred

```
public void messageTransferred(Message m,
                               DTNHost from,
                               DTNHost to,
                               boolean firstDelivery)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

m - The message that was transferred

`from` - Node where the message was transferred from
`to` - Node where the message was transferred to
`firstDelivery` - Was the target node final destination of the message and received this message for the first time.

newMessage

```
public void newMessage(Message m)
```

Description copied from interface: [MessageListener](#)

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

`m` - Message that was created

messageDeleted

```
public void messageDeleted(Message m,  
                           DTNHost where,  
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that was deleted
`where` - The host where the message was deleted
`dropped` - True if the message was dropped, false if removed

messageTransferAborted

```
public void messageTransferAborted(Message m,  
                                  DTNHost from,  
                                  DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that was being transferred
`from` - Node where the message was being transferred from
`to` - Node where the message was being transferred to

messageTransferStarted

```
public void messageTransferStarted(Message m,  
                                  DTNHost from,
```

[DTNHost](#) to)

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

- m - The message that is going to be transferred
- from - Node where the message is transferred from
- to - Node where the message is transferred to

done

public void **done**()

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement.map

Class DijkstraPathFinder

```
java.lang.Object
└─movement.map.DijkstraPathFinder
```

```
public class DijkstraPathFinder
extends java.lang.Object
```

Implementation of the Dijkstra's shortest path algorithm.

Constructor Summary

[DijkstraPathFinder\(int\[\] okMapNodes\)](#)

Constructor.

Method Summary

`java.util.List<MapNode>`

[getShortestPath\(MapNode from, MapNode to\)](#)

Finds and returns a shortest path between two map nodes

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

DijkstraPathFinder

```
public DijkstraPathFinder(int[] okMapNodes)
```

Constructor.

Parameters:

`okMapNodes` - The map node types that are OK for paths or null if all nodes are OK

Method Detail

getShortestPath

```
public java.util.List<MapNode> getShortestPath(MapNode from,
                                                MapNode to)
```

Finds and returns a shortest path between two map nodes

Parameters:

`from` - The source of the path

to - The destination of the path

Returns:

a shortest path between the source and destination nodes in a list of MapNodes or an empty list if such path is not available

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

routing

Class DirectDeliveryRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.DirectDeliveryRouter
```

```
public class DirectDeliveryRouter
extends ActiveRouter
```

Router that will deliver messages only to the final recipient.

Field Summary

Fields inherited from class routing.ActiveRouter

```
DELETE\_DELIVERED\_S, deleteDelivered, RESPONSE\_PREFIX, sendingConnections, TTL\_CHECK\_INTERVAL
```

Fields inherited from class routing.MessageRouter

```
B\_SIZE\_S, DENIED\_NO\_SPACE, DENIED\_OLD, DENIED\_TTL, DENIED\_UNSPECIFIED, MSG\_TTL\_S, msgTtl,
Q\_MODE\_FIFO, Q\_MODE\_RANDOM, RCV\_OK, SEND\_QUEUE\_MODE\_S, TRY\_LATER\_BUSY
```

Constructor Summary

protected	DirectDeliveryRouter (DirectDeliveryRouter r)
	DirectDeliveryRouter (Settings s)

Method Summary

DirectDeliveryRouter	replicate() Creates a replicate of this router.
void	update() Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.ActiveRouter

```
addToSendings, canStartTransfer, changedConnection, checkReceiving,
createNewMessage, dropExpiredMessages, exchangeDeliverableMessages, getConnections,
getMessagesForConnected, getOldestMessage, init, isSending, isTransferring,
makeRoomForMessage, makeRoomForNewMessage, messageTransferred, receiveMessage,
requestDeliverableMessages, shuffleMessages, startTransfer, transferAborted, transferDone,
tryAllMessages, tryAllMessagesToAllConnections, tryMessagesForConnected,
tryMessagesToConnections
```

Methods inherited from class routing.MessageRouter

```
addApplication, addToMessages, compareByQueueMode, deleteMessage, getApplications,  

getBufferSize, getFreeBufferSize, getHost, getMessage, getMessageCollection, getNrOfMessages,  

getRoutingInfo, hasMessage, isDeliveredMessage, isIncomingMessage, messageAborted,  

putToIncomingBuffer, removeFromIncomingBuffer, removeFromMessages, sendMessage,  

sortByQueueMode, toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail

DirectDeliveryRouter

```
public DirectDeliveryRouter(Settings s)
```

DirectDeliveryRouter

```
protected DirectDeliveryRouter(DirectDeliveryRouter r)
```

Method Detail

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

replicate

```
public DirectDeliveryRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class DistanceDelayReport

```
java.lang.Object
└ report.Report
  └ report.DistanceDelayReport
```

All Implemented Interfaces:[MessageListener](#)

```
public class DistanceDelayReport
extends Report
implements MessageListener
```

Report for how far apart the nodes were when the message was sent and how long time & how many hops it took to deliver it. Only messages created after the warm up period are counted. If message is not delivered, its delivery time & hop count are reported as -1

Field Summary

static java.lang.String

[SYNTAX](#)

Syntax of the report lines

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[DistanceDelayReport\(\)](#)

Constructor.

Method Summary

void	done() Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
protected void	init() Initializes the report output.
void	messageDeleted(Message m, DTNHost where, boolean dropped) Method is called when a message is deleted
void	messageTransferAborted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer was aborted before it finished
void	messageTransferred(Message m, DTNHost from, DTNHost to, boolean firstDelivery) This is called when a message is transferred between nodes
void	messageTransferStarted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer is started

void	newMessage (Message m)
------	---

This is called when a new message is created

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,  

getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,  

setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

SYNTAX

```
public static final java.lang.String SYNTAX
```

Syntax of the report lines

See Also:

[Constant Field Values](#)

Constructor Detail

DistanceDelayReport

```
public DistanceDelayReport()
```

Constructor.

Method Detail

init

```
protected void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

messageTransferred

```
public void messageTransferred(Message m,  

                               DTNHost from,  

                               DTNHost to,  

                               boolean firstDelivery)
```

This is called when a message is transferred between nodes

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

- m - The message that was transferred
 - from - Node where the message was transferred from
 - to - Node where the message was transferred to
 - firstDelivery - Was the target node final destination of the message and received this message for the first time.
-

newMessage

```
public void newMessage(Message m)
```

This is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

- m - Message that was created
-

messageDeleted

```
public void messageDeleted(Message m,  
                           DTNHost where,  
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

- m - The message that was deleted
 - where - The host where the message was deleted
 - dropped - True if the message was dropped, false if removed
-

messageTransferStarted

```
public void messageTransferStarted(Message m,  
                                  DTNHost from,  
                                  DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

- m - The message that is going to be transferred
 - from - Node where the message is transferred from
 - to - Node where the message is transferred to
-

messageTransferAborted

```
public void messageTransferAborted(Message m,  
                                 DTNHost from,  
                                 DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

m - The message that was being transferred
from - Node where the message was being transferred from
to - Node where the message was being transferred to

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class DTN2Events

```
java.lang.Object
└─input.DTN2Events
```

All Implemented Interfaces:

[EventQueue](#)

```
public class DTN2Events
extends java.lang.Object
implements EventQueue
```

Delivers bundles from dtnd to ONE. Must be configured as an external events generator in the configuration file.

Nested Class Summary

class	DTN2Events.ParserHandler
-------	--

Inner class that implements the CLA interface for receiving bundles from dtnd.

Constructor Summary

[DTN2Events\(Settings s\)](#)

Creates a new events object.

Method Summary

DTN2Events.ParserHandler	getParserHandler (int hostID, java.lang.String consoleHost, int consolePort)
--	--

Creates a parser handler for the given host.

ExternalEvent	nextEvent()
-------------------------------	-----------------------------

Returns the next event in the queue or ExternalEvent with time of Double.MAX_VALUE if there are no events left.

double	nextEventsTime()
--------	----------------------------------

Returns next event's time or Double.MAX_VALUE if there are no events left in the queue.

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#)

Constructor Detail

DTN2Events

```
public DTN2Events(Settings s)
```

Creates a new events object.

Parameters:

s - Settings

Method Detail

nextEvent

```
public ExternalEvent nextEvent()
```

Description copied from interface: [EventQueue](#)

Returns the next event in the queue or ExternalEvent with time of double.MAX_VALUE if there are no events left.

Specified by:

[nextEvent](#) in interface [EventQueue](#)

Returns:

The next event

nextEventsTime

```
public double nextEventsTime()
```

Description copied from interface: [EventQueue](#)

Returns next event's time or Double.MAX_VALUE if there are no events left in the queue.

Specified by:

[nextEventsTime](#) in interface [EventQueue](#)

Returns:

Next event's time

getParserHandler

```
public DTN2Events.ParserHandler getParserHandler(int hostID,
                                              java.lang.String consoleHost,
                                              int consolePort)
```

Creates a parser handler for the given host.

Parameters:

- hostID - ID of the host that this parser corresponds to
- consoleHost - Hostname of the dtnd
- consolePort - Console port of the dtnd

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Class DTN2Manager

```
java.lang.Object
└ core.DTN2Manager
```

```
public class DTN2Manager
extends java.lang.Object
```

Manages the external convergence layer connections to dtnd. Parses the configuration file and sets up the CLAParsers and EID->host mappings.

Nested Class Summary

static class	DTN2Manager.EIDHost
	EID to DTNHost mapping elements.

Constructor Summary

[DTN2Manager\(\)](#)

Method Summary

static void	addBundle(java.lang.String id, Bundle bundle) Stores a reference to a bundle corresponding to the given message.
static Bundle	getBundle(java.lang.String id) Returns the bundle associated with the given message id.
static DTN2Events	getEvents() Returns the DTN2Events object.
static java.util.Collection< DTN2Manager.EIDHost >	getHosts(java.lang.String EID) Returns a Collection of DTNHost objects corresponding to the given EID.
static CLAParser	getParser(DTNHost host) Returns the ECL parser associated with the host.
static DTN2Reporter	getReporter() Returns reference to the DTN2Reporter object.
static void	setEvents(DTN2Events events) Sets the DTN2Events object.
static void	setReporter(DTN2Reporter reporter) Sets the DTN2Reporter object used to pass messages from ONE to dtnd.
static void	setup(World world)

Sets up the dtnd connections by parsing the configuration file defined in the `DTN2.configFile` setting.

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`

Constructor Detail

DTN2Manager

```
public DTN2Manager()
```

Method Detail

setup

```
public static void setup(World world)
```

Sets up the dtnd connections by parsing the configuration file defined in the `DTN2.configFile` setting.

Parameters:

`world` - reference to the world that contains the nodes

setReporter

```
public static void setReporter(DTN2Reporter reporter)
```

Sets the `DTN2Reporter` object used to pass messages from ONE to dtnd. This should be used by the dynamically loaded `DTN2Reporter` object to allow other objects get reference to it.

Parameters:

`reporter` - the reporter object to save reference to

getReporter

```
public static DTN2Reporter getReporter()
```

Returns reference to the `DTN2Reporter` object.

Returns:

reference to the active `DTN2Reporter` object

setEvents

```
public static void setEvents(DTN2Events events)
```

Sets the `DTN2Events` object.

Parameters:

`events` - the active events object to use

getEvents

```
public static DTN2Events getEvents( )
```

Returns the DTN2Events object.

Returns:

the currently active events object.

getParser

```
public static CLAParser getParser(DTNHost host)
```

Returns the ECL parser associated with the host.

Parameters:

host - the host who's parser to return

Returns:

the host's parser.

getHosts

```
public static java.util.Collection<DTN2Manager.EIDHost> getHosts(java.lang.String EID)
```

Returns a Collection of DTNHost objects corresponding to the given EID.

Parameters:

EID - EID of the host

Returns:

the host corresponding to the EID

addBundle

```
public static void addBundle(java.lang.String id,  
                           Bundle bundle)
```

Stores a reference to a bundle corresponding to the given message.

Parameters:

id - the id of the message

bundle - the bundle associated with the message

getBundle

```
public static Bundle getBundle(java.lang.String id)
```

Returns the bundle associated with the given message id.

Parameters:

id - the message id

Returns:

the bundle associated with the message

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Class DTN2Manager.EIDHost

```
java.lang.Object
└ core.DTN2Manager.EIDHost
```

Enclosing class:[DTN2Manager](#)

```
public static class DTN2Manager.EIDHost
extends java.lang.Object
```

EID to DTNHost mapping elements.

Field Summary

java.lang.String	EID
DTNHost	host
int	host_id

Constructor Summary

```
DTN2Manager.EIDHost(java.lang.String eid, int host_id, DTNHost host)
```

Method Summary

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

EID

```
public java.lang.String EID
```

host_id

```
public int host_id
```

host

```
public DTNHost host
```

Constructor Detail**DTN2Manager.EIDHost**

```
public DTN2Manager.EIDHost(java.lang.String eid,  
                           int host_id,  
                           DTNHost host)
```

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class DTN2Reporter

```
java.lang.Object
  └ report.Report
    └ report.DTN2Reporter
```

All Implemented Interfaces:

[MessageListener](#)

```
public class DTN2Reporter
extends Report
implements MessageListener
```

The DTN2Reporter class is responsible for delivering bundles from The ONE to dtnd. To enable DTN2 connectivity, the class must be specified in the configuration file as a report class.

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[DTN2Reporter\(\)](#)

Creates a new reporter object.

Method Summary

void	messageDeleted (Message m, DTNHost where, boolean dropped)
	Method is called when a message is deleted
void	messageTransferAborted (Message m, DTNHost from, DTNHost to)
	Method is called when a message's transfer was aborted before it finished
void	messageTransferred (Message m, DTNHost from, DTNHost to, boolean firstDelivery)
	Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted (Message m, DTNHost from, DTNHost to)
	Method is called when a message's transfer is started
void	newMessage (Message m)
	Method is called when a new message is created

Methods inherited from class report.Report

```
addWarmupID, done, format, getAverage, getIntAverage, getIntMedian, getMedian,
getScenarioName, getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent,
removeWarmupID, setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail**DTN2Reporter**

```
public DTN2Reporter()
```

Creates a new reporter object.

Method Detail**newMessage**

```
public void newMessage(Message m)
```

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

m - Message that was created

messageTransferStarted

```
public void messageTransferStarted(Message m,
                                   DTNHost from,
                                   DTNHost to)
```

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

m - The message that is going to be transferred
from - Node where the message is transferred from
to - Node where the message is transferred to

messageDeleted

```
public void messageDeleted(Message m,
                           DTNHost where,
                           boolean dropped)
```

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

m - The message that was deleted

where - The host where the message was deleted
 dropped - True if the message was dropped, false if removed

messageTransferAborted

```
public void messageTransferAborted(Message m,
                                   DTNHost from,
                                   DTNHost to)
```

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

m - The message that was being transferred
 from - Node where the message was being transferred from
 to - Node where the message was being transferred to

messageTransferred

```
public void messageTransferred(Message m,
                               DTNHost from,
                               DTNHost to,
                               boolean firstDelivery)
```

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

m - The message that was transferred
 from - Node where the message was transferred from
 to - Node where the message was transferred to
 firstDelivery - Was the target node final destination of the message and received this message for the first time.

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Class DTNHost

```
java.lang.Object
└ core.DTNHost
```

All Implemented Interfaces:

java.lang.Comparable<[DTNHost](#)>

```
public class DTNHost
extends java.lang.Object
implements java.lang.Comparable<DTNHost>
```

A DTN capable host.

Constructor Summary

```
DTNHost(java.util.List<MessageListener> msgLs, java.util.List<MovementListener> movLs,
java.lang.String groupId, java.util.List<NetworkInterface> interf,
ModuleCommunicationBus comBus, MovementModel mmProto, MessageRouter mRouterProto)
```

Creates a new DTNHost.

Method Summary

<pre>int</pre>	compareTo(DTNHost h) Compares two DTNHosts by their addresses.
<pre>void</pre>	connect(DTNHost h) for tests only --- do not use!!!
<pre>void</pre>	connectionDown(Connection con)
<pre>void</pre>	connectionUp(Connection con) Informs the router of this host about state change in a connection object.
<pre>void</pre>	createNewMessage(Message m) Creates a new message to this host's router
<pre>void</pre>	deleteMessage(java.lang.String id, boolean drop) Deletes a message from this host
<pre>boolean</pre>	equals(DTNHost otherHost) Checks if a host is the same as this host by comparing the object reference
<pre>void</pre>	forceConnection(DTNHost anotherHost, java.lang.String interfaceId, boolean up) Force a connection event
<pre>int</pre>	getAddress() Returns the network-layer address of this host.
<pre>double</pre>	getBufferOccupancy()

		Returns the buffer occupancy percentage.
ModuleCommunicationBus		<code>getComBus()</code> Returns this host's ModuleCommunicationBus
java.util.List<Connection>		<code>getConnections()</code> Returns a copy of the list of connections this host has with other hosts
protected NetworkInterface		<code>getInterface(int interfaceNo)</code> Find the network interface based on the index
protected NetworkInterface		<code>getInterface(java.lang.String interfacetype)</code> Find the network interface based on the interfacetype
java.util.List<NetworkInterface>		<code>getInterfaces()</code> Returns the interface objects of the node
Coord		<code>getLocation()</code> Returns the current location of this host.
java.util.Collection<Message>		<code>getMessageCollection()</code> Returns the messages in a collection.
int		<code>getNrofMessages()</code> Returns the number of messages this node is carrying.
Path		<code>getPath()</code> Returns the Path this node is currently traveling or null if no path is in use at the moment.
MessageRouter		<code>getRouter()</code> Returns the router of this host
RoutingInfo		<code>getRoutingInfo()</code> Returns routing info of this host's router.
boolean		<code>isActive()</code> Returns true if this node is active (false if not)
void		<code>messageAborted(java.lang.String id, DTNHost from, int bytesRemaining)</code> Informs the host that a message transfer was aborted.
void		<code>messageTransferred(java.lang.String id, DTNHost from)</code> Informs the host that a message was successfully transferred.
void		<code>move(double timeIncrement)</code> Moves the node towards the next waypoint or waits if it is not time to move yet
int		<code>receiveMessage(Message m, DTNHost from)</code> Start receiving a message from another host
boolean		<code>requestDeliverableMessages(Connection con)</code> Requests for deliverable message from this host to be sent through a connection.
static void		<code>reset()</code> Reset the host and its interfaces
void		<code>sendMessage(java.lang.String id, DTNHost to)</code> Sends a message from this host to another host
void		<code>setLocation(Coord location)</code> Sets the Node's location overriding any location set by movement model
void		<code>setName(java.lang.String name)</code>

	Sets the Node's name overriding the default name (groupId + netAddress)
java.lang.String	toString() Returns a string presentation of the host.
void	update(boolean simulateConnections) Updates node's network layer and router.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Constructor Detail

DTNHost

```
public DTNHost(java.util.List<MessageListener> msgLs,
               java.util.List<MovementListener> movLs,
               java.lang.String groupId,
               java.util.List<NetworkInterface> interf,
               ModuleCommunicationBus comBus,
               MovementModel mmProto,
               MessageRouter mRouterProto)
```

Creates a new DTNHost.

Parameters:

- msgLs - Message listeners
- movLs - Movement listeners
- groupId - GroupID of this host
- interf - List of NetworkInterfaces for the class
- comBus - Module communication bus object
- mmProto - Prototype of the movement model of this host
- mRouterProto - Prototype of the message router of this host

Method Detail

reset

```
public static void reset()
```

Reset the host and its interfaces

isActive

```
public boolean isActive()
```

Returns true if this node is active (false if not)

Returns:

true if this node is active (false if not)

getRouter

```
public MessageRouter getRouter()
```

Returns the router of this host

Returns:

the router of this host

getAddress

```
public int getAddress()
```

Returns the network-layer address of this host.

getComBus

```
public ModuleCommunicationBus getComBus()
```

Returns this hosts's ModuleCommunicationBus

Returns:

this hosts's ModuleCommunicationBus

connectionUp

```
public void connectionUp(Connection con)
```

Informs the router of this host about state change in a connection object.

Parameters:

con - The connection object whose state changed

connectionDown

```
public void connectionDown(Connection con)
```

getConnections

```
public java.util.List<Connection> getConnections()
```

Returns a copy of the list of connections this host has with other hosts

Returns:

a copy of the list of connections this host has with other hosts

getLocation

```
public Coord getLocation()
```

Returns the current location of this host.

Returns:

The location

getPath

```
public Path getPath()
```

Returns the Path this node is currently traveling or null if no path is in use at the moment.

Returns:

The path this node is traveling

setLocation

```
public void setLocation(Coord location)
```

Sets the Node's location overriding any location set by movement model

Parameters:

location - The location to set

setName

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

Sets the Node's name overriding the default name (groupId + netAddress)

Parameters:

name - The name to set

getMessageCollection

```
public java.util.Collection<Message> getMessageCollection()
```

Returns the messages in a collection.

Returns:

Messages in a collection

getNrofMessages

```
public int getNrofMessages()
```

Returns the number of messages this node is carrying.

Returns:

How many messages the node is carrying currently.

getBufferOccupancy

```
public double getBufferOccupancy()
```

Returns the buffer occupancy percentage. Occupancy is 0 for empty buffer but can be over 100 if a created message is bigger than buffer space that could be freed.

Returns:

Buffer occupancy percentage

getRoutingInfo

```
public RoutingInfo getRoutingInfo()
```

Returns routing info of this host's router.

Returns:

The routing info.

getInterfaces

```
public java.util.List<NetworkInterface> getInterfaces()
```

Returns the interface objects of the node

getInterface

```
protected NetworkInterface getInterface(int interfaceNo)
```

Find the network interface based on the index

getInterface

```
protected NetworkInterface getInterface(java.lang.String interfacetype)
```

Find the network interface based on the interfacetype

forceConnection

```
public void forceConnection(DTNHost anotherHost,  
                           java.lang.String interfaceId,  
                           boolean up)
```

Force a connection event

connect

```
public void connect(DTNHost h)
```

for tests only --- do not use!!!

update

```
public void update(boolean simulateConnections)
```

Updates node's network layer and router.

Parameters:

simulateConnections - Should network layer be updated too

move

```
public void move(double timeIncrement)
```

Moves the node towards the next waypoint or waits if it is not time to move yet

Parameters:

`timeIncrement` - How long time the node moves

sendMessage

```
public void sendMessage(java.lang.String id,
                      DTNHost to)
```

Sends a message from this host to another host

Parameters:

`id` - Identifier of the message
`to` - Host the message should be sent to

receiveMessage

```
public int receiveMessage(Message m,
                         DTNHost from)
```

Start receiving a message from another host

Parameters:

`m` - The message
`from` - Who the message is from

Returns:

The value returned by [MessageRouter.receiveMessage\(Message, DTNHost\)](#)

requestDeliverableMessages

```
public boolean requestDeliverableMessages(Connection con)
```

Requests for deliverable message from this host to be sent trough a connection.

Parameters:

`con` - The connection to send the messages trough

Returns:

True if this host started a transfer, false if not

messageTransferred

```
public void messageTransferred(java.lang.String id,
                               DTNHost from)
```

Informs the host that a message was successfully transferred.

Parameters:

`id` - Identifier of the message
`from` - From who the message was from

messageAborted

```
public void messageAborted(java.lang.String id,
                           DTNHost from,
                           int bytesRemaining)
```

Informs the host that a message transfer was aborted.

Parameters:

- id - Identifier of the message
 - from - From who the message was from
 - bytesRemaining - Number of bytes that were left before the transfer would have been ready; or -1 if the number of bytes is not known
-

createNewMessage

```
public void createNewMessage(Message m)
```

Creates a new message to this host's router

Parameters:

- m - The message to create
-

deleteMessage

```
public void deleteMessage(java.lang.String id,
                          boolean drop)
```

Deletes a message from this host

Parameters:

- id - Identifier of the message
 - drop - True if the message is deleted because of "dropping" (e.g. buffer is full) or false if it was deleted for some other reason (e.g. the message got delivered to final destination). This effects the way the removing is reported to the message listeners.
-

toString

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

Returns a string presentation of the host.

Overrides:

`toString` in class `java.lang.Object`

Returns:

Host's name

equals

```
public boolean equals(DTNHost otherHost)
```

Checks if a host is the same as this host by comparing the object reference

Parameters:

otherHost

- The other host

Returns:

True if the hosts objects are the same object

compareTo

public int compareTo([DTNHost](#) h)

Compares two DTNHosts by their addresses.

Specified by:

compareTo in interface [java.lang.Comparable<DTNHost>](#)

See Also:

[Comparable.compareTo\(Object\)](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Class DTNSim

```
java.lang.Object
└ core.DTNSim
```

```
public class DTNSim
extends java.lang.Object
```

Simulator's main class

Field Summary

static java.lang.String	BATCH_MODE_FLAG	If this option ("-b") is given to program, batch mode and Text UI are used
static java.lang.String	RANGE_DELIMETER	Delimiter for batch mode index range values (colon)
static java.lang.String	RESET_METHOD_NAME	Name of the static method that all resettable classes must have

Constructor Summary

[DTNSim\(\)](#)

Method Summary

static void	main(java.lang.String[] args)	Starts the user interface with given arguments.
static void	registerForReset(java.lang.String className)	Registers a class for resetting.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

BATCH_MODE_FLAG

```
public static final java.lang.String BATCH_MODE_FLAG
```

If this option ("-b") is given to program, batch mode and Text UI are used

See Also:

[Constant Field Values](#)

RANGE_DELIMETER

```
public static final java.lang.String RANGE_DELIMETER
```

Delimiter for batch mode index range values (colon)

See Also:

[Constant Field Values](#)

RESET_METHOD_NAME

```
public static final java.lang.String RESET_METHOD_NAME
```

Name of the static method that all resettable classes must have

See Also:

[registerForReset\(String\)](#), [Constant Field Values](#)

Constructor Detail

DTNSim

```
public DTNSim()
```

Method Detail

main

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

Starts the user interface with given arguments. If first argument is [BATCH_MODE_FLAG](#), the batch mode and text UI is started. The batch mode option must be followed by the number of runs, or a with a combination of starting run and the number of runs, delimited with a [":"](#). Different settings from run arrays are used for different runs (see [Settings.setRunIndex\(int\)](#)). Following arguments are the settings files for the simulation run (if any). For GUI mode, the number before settings files (if given) is the run index to use for that run.

Parameters:

args - Command line arguments

registerForReset

```
public static void registerForReset(java.lang.String className)
```

Registers a class for resetting. Reset is performed after every batch run of the simulator to reset the class' state to initial state. All classes that have static fields that should be resetted to initial values between the batch runs should register using this method. The given class must have a static implementation for the resetting method (a method called ["reset"](#) without any parameters).

Parameters:

className - Full name (i.e., containing the packet path) of the class to register. For example:
core.SimClock

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

gui

Class DTNSimGUI

```
java.lang.Object
  ↘ ui.DTNSimUI
    ↘ gui.DTNSimGUI
```

```
public class DTNSimGUI
extends DTNSimUI
```

Graphical User Interface for simulator

Field Summary

Fields inherited from class ui.DTNSimUI

```
lastUpdate, MM_WARMUP_S, NROF_REPORT_S, REPORT_S, reports, scen, simCancelled, simDone, world
```

Constructor Summary

```
DTNSimGUI()
```

Method Summary

void	centerViewAt(Coord loc) Sets certain location to be in the center of the playfield view
void	closeSim() Closes the program if simulation is done or cancels it.
Coord	getCenterViewCoord() Returns the world coordinates that are currently in the center of the viewport
InfoPanel	 getInfoPanel() Returns the info panel of the GUI
MainWindow	getParentFrame() Returns the parent frame (window) of the gui.
protected void	runSim() Runs simulation after the model has been initialized.
void	setFocus(DTNHost host) Sets a node's graphical presentation in the center of the playfield view
void	setPaused(boolean paused) Sets the pause of the simulation on/off
void	showPath(Path path) Shows a path on the playfield
void	update(boolean forcedUpdate) Updates the GUI

Methods inherited from class ui.DTNSimUI[addReport](#), [done](#), [start](#)**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail**DTNSimGUI**

```
public DTNSimGUI()
```

Method Detail**runSim**

```
protected void runSim()
```

Description copied from class: [DTNSimUI](#)

Runs simulation after the model has been initialized.

Specified by:

[runSim](#) in class [DTNSimUI](#)

closeSim

```
public void closeSim()
```

Closes the program if simulation is done or cancels it.

update

```
public void update(boolean forcedUpdate)
```

Updates the GUI

setPaused

```
public void setPaused(boolean paused)
```

Sets the pause of the simulation on/off

Parameters:

paused - True if pause should be set on

setFocus

```
public void setFocus(DTNHost host)
```

Sets a node's graphical presentation in the center of the playfield view

Parameters:

host - The node to center

showPath

```
public void showPath(Path path)
```

Shows a path on the playfield

Parameters:

path - The path to show

getCenterViewCoord

```
public Coord getCenterViewCoord()
```

Returns the world coordinates that are currently in the center of the viewport

Returns:

The coordinates

centerViewAt

```
public void centerViewAt(Coord loc)
```

Sets certain location to be in the center of the playfield view

Parameters:

loc - The location to center

getInfoPanel

```
public InfoPanel getInfoPanel()
```

Returns the info panel of the GUI

Returns:

the info panel of the GUI

getParentFrame

```
public MainWindow getParentFrame()
```

Returns the parent frame (window) of the gui.

Returns:

The parent frame

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)PREV CLASS [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

ui

Class DTNSimTextUI

java.lang.Object

```

└ ui.DTNSimUI
    └ ui.DTNSimTextUI

```

```
public class DTNSimTextUI
extends DTNSimUI
```

Simple text-based user interface.

Field Summary

static long

[UI_UP_INTERVAL](#)

How often the UI view is updated (milliseconds)

Fields inherited from class ui.DTNSimUI

[lastUpdate](#), [MM_WARMUP_S](#), [NROF_REPORT_S](#), [REPORT_S](#), [reports](#), [scen](#), [simCancelled](#), [simDone](#), [world](#)

Constructor Summary

[DTNSimTextUI\(\)](#)

Method Summary

protected void

[runSim\(\)](#)

Runs simulation after the model has been initialized.

Methods inherited from class ui.DTNSimUI

[addReport](#), [done](#), [start](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Field Detail

UI_UP_INTERVAL

public static final long [UI_UP_INTERVAL](#)

How often the UI view is updated (milliseconds)

See Also:

[Constant Field Values](#)

Constructor Detail

DTNSimTextUI

```
public DTNSimTextUI()
```

Method Detail

runSim

```
protected void runSim()
```

Description copied from class: [DTNSimUI](#)

Runs simulation after the model has been initialized.

Specified by:

[runSim](#) in class [DTNSimUI](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

ui

Class DTNSimUI

```
java.lang.Object
└ ui.DTNSimUI
```

Direct Known Subclasses:[DTNSimGUI](#), [DTNSimTextUI](#)

```
public abstract class DTNSimUI
extends java.lang.Object
```

Abstract superclass for user interfaces; contains also some simulation settings.

Field Summary

protected double	lastUpdate simtime of last UI update
static java.lang.String	MM_WARMUP_S Movement model warmup time -setting id ("MovementModel.warmup").
static java.lang.String	NROF_REPORT_S Number of reports -setting id ("Report.nrofReports").
static java.lang.String	REPORT_S Report class name -setting id prefix ("Report.report").
protected java.util.Vector< Report >	reports Reports that are loaded for this simulation
protected SimScenario	scen Scenario of the current simulation
protected boolean	simCancelled is simulation termination requested
protected boolean	simDone has simulation terminated normally
protected World	world The World where all actors of the simulator are

Constructor Summary

[DTNSimUI\(\)](#)

Constructor.

Method Summary

protected void	addReport(Report r) Adds a new report for simulator
void	done()

	Runs maintenance jobs that are needed before exiting.
protected abstract void	runSim() Runs simulation after the model has been initialized.
void	start() Starts the simulation.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Field Detail**NROF_REPORT_S**public static final java.lang.String **NROF_REPORT_S**

Number of reports -setting id ("Report.nrofReports"). Defines how many reports are loaded.

See Also:[Constant Field Values](#)**REPORT_S**public static final java.lang.String **REPORT_S**

Report class name -setting id prefix ("Report.report"). Defines name(s) of the report classes to load. Must be suffixed with numbers starting from one.

See Also:[Constant Field Values](#)**MM_WARMUP_S**public static final java.lang.String **MM_WARMUP_S**

Movement model warmup time -setting id ("MovementModel.warmup"). Defines how many seconds of movement simulation is run without connectivity etc. checks before starting the real simulation.

See Also:[Constant Field Values](#)**world**protected [world](#) **world**

The World where all actors of the simulator are

reportsprotected java.util.Vector<[Report](#)> **reports**

Reports that are loaded for this simulation

simDone

protected boolean **simDone**

has simulation terminated normally

simCancelled

protected boolean **simCancelled**

is simulation termination requested

scen

protected [SimScenario](#) **scen**

Scenario of the current simulation

lastUpdate

protected double **lastUpdate**

simtime of last UI update

Constructor Detail

DTNSimUI

public **DTNSimUI()**

Constructor.

Method Detail

start

public void **start()**

Starts the simulation.

runSim

protected abstract void **runSim()**

Runs simulation after the model has been initialized.

done

```
public void done()
```

Runs maintenance jobs that are needed before exiting.

addReport

```
protected void addReport(Report r)
```

Adds a new report for simulator

Parameters:

r - Report to add

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class EncountersVSUniqueEncountersReport

```
java.lang.Object
└ report.Report
  └ report.EncountersVSUniqueEncountersReport
```

All Implemented Interfaces:[ConnectionListener](#), [UpdateListener](#)

```
public class EncountersVSUniqueEncountersReport
extends Report
implements ConnectionListener, UpdateListener
```

The total- vs. the unique encounters for each node

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[EncountersVSUniqueEncountersReport\(\)](#)

Method Summary

void	done()	Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
void	hostsConnected(DTNHost host1, DTNHost host2)	Method is called when two hosts are connected.
void	hostsDisconnected(DTNHost host1, DTNHost host2)	Method is called when connection between hosts is disconnected.
void	updated(java.util.List<DTNHost> hosts)	Method is called on every update cycle.

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent, removeWarmupID,
setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

Constructor Detail

EncountersVSUniqueEncountersReport

```
public EncountersVSUniqueEncountersReport()
```

Method Detail

hostsConnected

```
public void hostsConnected(DTNHost host1,
                           DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the connection

host2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost host1,
                             DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

updated

```
public void updated(java.util.List<DTNHost> hosts)
```

Description copied from interface: [UpdateListener](#)

Method is called on every update cycle.

Specified by:

[updated](#) in interface [UpdateListener](#)

Parameters:

hosts - A list of all hosts in the world

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class EnergyAwareRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.EnergyAwareRouter
```

All Implemented Interfaces:[ModuleCommunicationListener](#)

```
public class EnergyAwareRouter
extends ActiveRouter
implements ModuleCommunicationListener
```

Energy level-aware variant of Epidemic router.

Field Summary

static java.lang.String	ENERGY VALUE ID ModuleCommunicationBus identifier for the "current amount of energy left" variable.
static java.lang.String	INIT ENERGY S Initial units of energy -setting id ("intialEnergy").
static java.lang.String	SCAN ENERGY S Energy usage per scanning -setting id ("scanEnergy").
static java.lang.String	TRANSMIT ENERGY S Energy usage per second when sending -setting id ("transmitEnergy").
static java.lang.String	WARMUP S Energy update warmup period -setting id ("energyWarmup").

Fields inherited from class routing.ActiveRouter

[DELETE DELIVERED S](#), [deleteDelivered](#), [RESPONSE PREFIX](#), [sendingConnections](#), [TTL CHECK INTERVAL](#)

Fields inherited from class routing.MessageRouter

[B SIZE S](#), [DENIED NO SPACE](#), [DENIED OLD](#), [DENIED TTL](#), [DENIED UNSPECIFIED](#), [MSG TTL S](#), [msgTtl](#), [Q MODE FIFO](#), [Q MODE RANDOM](#), [RCV OK](#), [SEND QUEUE MODE S](#), [TRY LATER BUSY](#)

Constructor Summary

protected	EnergyAwareRouter (EnergyAwareRouter r) Copy constructor.
	EnergyAwareRouter (Settings s) Constructor.

Method Summary

protected int	checkReceiving(Message m) Checks if router "wants" to start receiving message (i.e.
void	moduleValueChanged(java.lang.String key, java.lang.Object newValue) Called by the combus is the energy value is changed
protected void	reduceEnergy(double amount) Updates the current energy so that the given amount is reduced from it.
protected void	reduceSendingAndScanningEnergy() Reduces the energy reserve for the amount that is used by sending data and scanning for the other nodes.
EnergyAwareRouter	replicate() Creates a replicate of this router.
protected void	setEnergy(double[] range) Sets the current energy level into the given range using uniform random distribution.
java.lang.String	toString() Returns a String presentation of this router
void	update() Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.[ActiveRouter](#)

[addToSendConnections](#), [canStartTransfer](#), [changedConnection](#), [createNewMessage](#), [dropExpiredMessages](#), [exchangeDeliverableMessages](#), [getConnections](#), [getMessagesForConnected](#), [getOldestMessage](#), [init](#), [isSending](#), [isTransferring](#), [makeRoomForMessage](#), [makeRoomForNewMessage](#), [messageTransferred](#), [receiveMessage](#), [requestDeliverableMessages](#), [shuffleMessages](#), [startTransfer](#), [transferAborted](#), [transferDone](#), [tryAllMessages](#), [tryAllMessagesToAllConnections](#), [tryMessagesForConnected](#), [tryMessagesToConnections](#)

Methods inherited from class routing.[MessageRouter](#)

[addApplication](#), [addToMessages](#), [compareByQueueMode](#), [deleteMessage](#), [getApplications](#), [getBufferSize](#), [getFreeBufferSize](#), [getHost](#), [getMessage](#), [getMessageCollection](#), [getNrofMessages](#), [getRoutingInfo](#), [hasMessage](#), [isDeliveredMessage](#), [isIncomingMessage](#), [messageAborted](#), [putToIncomingBuffer](#), [removeFromIncomingBuffer](#), [removeFromMessages](#), [sendMessage](#), [sortByQueueMode](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

INIT_ENERGY_S

public static final java.lang.String **INIT_ENERGY_S**

Initial units of energy -setting id ("intialEnergy"). Can be either a single value, or a range of two values. In the latter case, the used value is a uniformly distributed random value between the two values.

See Also:

[Constant Field Values](#)

SCAN_ENERGY_S

public static final java.lang.String **SCAN_ENERGY_S**

Energy usage per scanning -setting id ("scanEnergy").

See Also:

[Constant Field Values](#)

TRANSMIT_ENERGY_S

```
public static final java.lang.String TRANSMIT_ENERGY_S
```

Energy usage per second when sending -setting id ("transmitEnergy").

See Also:

[Constant Field Values](#)

WARMUP_S

```
public static final java.lang.String WARMUP_S
```

Energy update warmup period -setting id ("energyWarmup"). Defines the simulation time after which the energy level starts to decrease due to scanning, transmissions, etc. Default value = 0. If value of "-1" is defined, uses the value from the report warmup setting [Report.WARMUP_S](#) from the namespace "[Report](#)".

See Also:

[Constant Field Values](#)

ENERGY_VALUE_ID

```
public static final java.lang.String ENERGY_VALUE_ID
```

[ModuleCommunicationBus](#) identifier for the "current amount of energy left" variable. Value type: double

See Also:

[Constant Field Values](#)

Constructor Detail

EnergyAwareRouter

```
public EnergyAwareRouter(Settings s)
```

Constructor. Creates a new message router based on the settings in the given Settings object.

Parameters:

s - The settings object

EnergyAwareRouter

```
protected EnergyAwareRouter(EnergyAwareRouter r)
```

Copy constructor.

Parameters:

r - The router prototype where setting values are copied from

Method Detail

setEnergy

```
protected void setEnergy(double[] range)
```

Sets the current energy level into the given range using uniform random distribution.

Parameters:

range - The min and max values of the range, or if only one value is given, that is used as the energy level

checkReceiving

```
protected int checkReceiving(Message m)
```

Description copied from class: [ActiveRouter](#)

Checks if router "wants" to start receiving message (i.e. router isn't transferring, doesn't have the message and has room for it).

Overrides:

[checkReceiving](#) in class [ActiveRouter](#)

Parameters:

m - The message to check

Returns:

A return code similar to [MessageRouter.receiveMessage\(Message, DTNHost\)](#), i.e. [MessageRouter.RCV_OK](#) if receiving seems to be OK, TRY_LATER_BUSY if router is transferring, DENIED_OLD if the router is already carrying the message or it has been delivered to this router (as final recipient), or DENIED_NO_SPACE if the message does not fit into buffer

reduceEnergy

```
protected void reduceEnergy(double amount)
```

Updates the current energy so that the given amount is reduced from it. If the energy level goes below zero, sets the level to zero. Does nothing if the warmup time has not passed.

Parameters:

amount - The amount of energy to reduce

reduceSendingAndScanningEnergy

```
protected void reduceSendingAndScanningEnergy()
```

Reduces the energy reserve for the amount that is used by sending data and scanning for the other nodes.

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

replicate

```
public EnergyAwareRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

moduleValueChanged

```
public void moduleValueChanged(java.lang.String key,
                               java.lang.Object newValue)
```

Called by the combus is the energy value is changed

Specified by:

[moduleValueChanged](#) in interface [ModuleCommunicationListener](#)

Parameters:

key - The energy ID
newValue - The new energy value

toString

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

Description copied from class: [MessageRouter](#)

Returns a String presentation of this router

Overrides:

[toString](#) in class [MessageRouter](#)

Returns:

A String presentation of this router

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class EnergyLevelReport

```
java.lang.Object
└ report.Report
  └ report.EnergyLevelReport
```

All Implemented Interfaces:

[UpdateListener](#)

```
public class EnergyLevelReport
extends Report
implements UpdateListener
```

Node energy level report. Reports the energy level of all (or only some) nodes every configurable-amount-of seconds. Writes reports only after the warmup period.

Field Summary

protected int	granularity value of the granularity setting
static java.lang.String	GRANULARITY Reporting granularity -setting id ("granularity").
protected double	lastUpdate time of last update
static java.lang.String	REPORTED NODES Optional reported nodes (comma separated list of network addresses).
protected java.util.HashSet<java.lang.Integer>	reportedNodes Networks addresses (integers) of the nodes which are reported

Fields inherited from class report.Report

```
DEF PRECISION, INTERVAL_SETTING, INTERVALLED FORMAT, NAN, out, OUT SUFFIX, OUTPUT SETTING,  
PRECISION SETTING, REPORT NS, REPORTDIR SETTING, WARMUP S, warmupIDs, warmupTime
```

Constructor Summary

[EnergyLevelReport\(\)](#)

Constructor.

Method Summary

void [updated](#)(java.util.List<[DTNHost](#)> hosts)

Creates a new snapshot of the energy levels if "granularity" seconds have passed since the last snapshot.

Methods inherited from class report.Report

```
addWarmupID, done, format, getAverage, getIntAverage, getIntMedian, getMedian,  

getScenarioName, getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent,  

removeWarmupID, setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail**GRANULARITY**

```
public static final java.lang.String GRANULARITY
```

Reporting granularity -setting id ("granularity"). Defines the interval how often (seconds) a new snapshot of energy levels is created

See Also:

[Constant Field Values](#)

REPORTED_NODES

```
public static final java.lang.String REPORTED_NODES
```

Optional reported nodes (comma separated list of network addresses). By default all nodes are reported.

See Also:

[Constant Field Values](#)

granularity

```
protected final int granularity
```

value of the granularity setting

lastUpdate

```
protected double lastUpdate
```

time of last update

reportedNodes

```
protected java.util.HashSet<java.lang.Integer> reportedNodes
```

Networks addresses (integers) of the nodes which are reported

Constructor Detail**EnergyLevelReport**

```
public EnergyLevelReport()
```

Constructor. Reads the settings and initializes the report module.

Method Detail

updated

```
public void updated(java.util.List<DTNHost> hosts)
```

Creates a new snapshot of the energy levels if "granularity" seconds have passed since the last snapshot.

Specified by:

[updated](#) in interface [UpdateListener](#)

Parameters:

hosts - All the hosts in the world

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class EpidemicOracleRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.EpidemicOracleRouter
```

```
public class EpidemicOracleRouter
extends ActiveRouter
```

Epidemic message router with an oracle that tells when a message is delivered and that message is then removed from all nodes that use this routing module. This router also **ignores message size and all messages are delivered immediately**.

Note: This router module also bypasses ActiveRouter.update()

Field Summary

Fields inherited from class routing.ActiveRouter

```
DELETE_DELIVERED_S, deleteDelivered, RESPONSE_PREFIX, sendingConnections, TTL_CHECK_INTERVAL
```

Fields inherited from class routing.MessageRouter

```
B_SIZE_S, DENIED_NO_SPACE, DENIED_OLD, DENIED_TTL, DENIED_UNSPECIFIED, MSG_TTL_S, msgTtl,
Q_MODE_FIFO, Q_MODE_RANDOM, RCV_OK, SEND_QUEUE_MODE_S, TRY_LATER_BUSY
```

Constructor Summary

protected	EpidemicOracleRouter (EpidemicOracleRouter r) Copy constructor.
-----------	---

	EpidemicOracleRouter (Settings s) Constructor.
--	--

Method Summary

void	changedConnection (Connection con) Called when a connection's state changes.
protected int	checkReceiving (Message m) Checks if router "wants" to start receiving message (i.e.
boolean	createNewMessage (Message m) Creates a new message to the router.
Message	messageTransferred (java.lang.String id, DTNHost from) This method should be called (on the receiving host) after a message was successfully transferred.
void	removeDeliveredMessage (java.lang.String id)

	Removes the message with the given ID from this router, if the router has that message; otherwise does nothing.
EpidemicOracleRouter	replicate() Creates a replicate of this router.
static void	reset() Resets the static router list
protected void	transferDone(Connection con) Method is called just before a transfer is finalized at ActiveRouter.update() .
void	update() Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.[ActiveRouter](#)

[addToSendingConnections](#), [canStartTransfer](#), [dropExpiredMessages](#), [exchangeDeliverableMessages](#), [getConnections](#), [getMessagesForConnected](#), [getOldestMessage](#), [init](#), [isSending](#), [isTransferring](#), [makeRoomForMessage](#), [makeRoomForNewMessage](#), [receiveMessage](#), [requestDeliverableMessages](#), [shuffleMessages](#), [startTransfer](#), [transferAborted](#), [tryAllMessages](#), [tryAllMessagesToAllConnections](#), [tryMessagesForConnected](#), [tryMessagesToConnections](#)

Methods inherited from class routing.[MessageRouter](#)

[addApplication](#), [addToMessages](#), [compareByQueueMode](#), [deleteMessage](#), [getApplications](#), [getBufferSize](#), [getFreeBufferSize](#), [getHost](#), [getMessage](#), [getMessageCollection](#), [getNrofMessages](#), [getRoutingInfo](#), [hasMessage](#), [isDeliveredMessage](#), [isIncomingMessage](#), [messageAborted](#), [putToIncomingBuffer](#), [removeFromIncomingBuffer](#), [removeFromMessages](#), [sendMessage](#), [sortByQueueMode](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

EpidemicOracleRouter

```
public EpidemicOracleRouter(Settings s)
```

Constructor. Creates a new message router based on the settings in the given Settings object.

Parameters:

s - The settings object

EpidemicOracleRouter

```
protected EpidemicOracleRouter(EpidemicOracleRouter r)
```

Copy constructor.

Parameters:

r - The router prototype where setting values are copied from

Method Detail

changedConnection

```
public void changedConnection(Connection con)
```

Description copied from class: [ActiveRouter](#)

Called when a connection's state changes. This version doesn't do anything but subclasses may want to override this.

Overrides:

[changedConnection](#) in class [ActiveRouter](#)

Parameters:

con - The connection that changed

createNewMessage

```
public boolean createNewMessage(Message m)
```

Description copied from class: [MessageRouter](#)

Creates a new message to the router.

Overrides:

[createNewMessage](#) in class [ActiveRouter](#)

Parameters:

m - The message to create

Returns:

True if the creation succeeded, false if not (e.g. the message was too big for the buffer)

removeDeliveredMessage

```
public void removeDeliveredMessage(java.lang.String id)
```

Removes the message with the given ID from this router, if the router has that message; otherwise does nothing. If the router was transferring the message, the transfer is aborted.

Parameters:

id - ID of the message to be removed

messageTransferred

```
public Message messageTransferred(java.lang.String id,
                                    DTNHost from)
```

Description copied from class: [MessageRouter](#)

This method should be called (on the receiving host) after a message was successfully transferred. The transferred message is put to the message buffer unless this host is the final recipient of the message.

Overrides:

[messageTransferred](#) in class [ActiveRouter](#)

Parameters:

id - Id of the transferred message

from - Host the message was from (previous hop)

Returns:

The message that this host received

checkReceiving

```
protected int checkReceiving(Message m)
```

Description copied from class: [ActiveRouter](#)

Checks if router "wants" to start receiving message (i.e. router isn't transferring, doesn't have the message and has room for it).

Overrides:

[checkReceiving](#) in class [ActiveRouter](#)

Parameters:

m - The message to check

Returns:

A return code similar to [MessageRouter.receiveMessage\(Message, DTNHost\)](#), i.e. [MessageRouter.RCV_OK](#) if receiving seems to be OK, TRY_LATER_BUSY if router is transferring, DENIED_OLD if the router is already carrying the message or it has been delivered to this router (as final recipient), or DENIED_NO_SPACE if the message does not fit into buffer

transferDone

```
protected void transferDone(Connection con)
```

Description copied from class: [ActiveRouter](#)

Method is called just before a transfer is finalized at [ActiveRouter.update\(\)](#). Subclasses that are interested of the event may want to override this.

Overrides:

[transferDone](#) in class [ActiveRouter](#)

Parameters:

con - The connection whose transfer was finalized

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

replicate

```
public EpidemicOracleRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

reset

public static void **reset()**

Resets the static router list

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

routing

Class EpidemicRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.EpidemicRouter
```

```
public class EpidemicRouter
extends ActiveRouter
```

Epidemic message router with drop-oldest buffer and only single transferring connections at a time.

Field Summary

Fields inherited from class routing.ActiveRouter

[DELETE_DELIVERED_S](#), [deleteDelivered](#), [RESPONSE_PREFIX](#), [sendingConnections](#), [TTL_CHECK_INTERVAL](#)

Fields inherited from class routing.MessageRouter

[B_SIZE_S](#), [DENIED_NO_SPACE](#), [DENIED_OLD](#), [DENIED_TTL](#), [DENIED_UNSPECIFIED](#), [MSG_TTL_S](#), [msgTtl](#), [Q_MODE_FIFO](#), [Q_MODE_RANDOM](#), [RCV_OK](#), [SEND_QUEUE_MODE_S](#), [TRY_LATER_BUSY](#)

Constructor Summary

protected	EpidemicRouter(EpidemicRouter r)
-----------	--

Copy constructor.

	EpidemicRouter(Settings s)
--	--

Constructor.

Method Summary

[EpidemicRouter](#).[replicate\(\)](#)

Creates a replicate of this router.

void	update()
------	--------------------------

Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.ActiveRouter

```
addToSendings, canStartTransfer, changedConnection, checkReceiving,
createNewMessage, dropExpiredMessages, exchangeDeliverableMessages, getConnections,
getMessagesForConnected, getOldestMessage, init, isSending, isTransferring,
makeRoomForMessage, makeRoomForNewMessage, messageTransferred, receiveMessage,
requestDeliverableMessages, shuffleMessages, startTransfer, transferAborted, transferDone,
tryAllMessages, tryAllMessagesToAllConnections, tryMessagesForConnected,
tryMessagesToConnections
```

Methods inherited from class routing.MessageRouter

```
addApplication, addToMessages, compareByQueueMode, deleteMessage, getApplications,  

getBufferSize, getFreeBufferSize, getHost, getMessage, getMessageCollection, getNrOfMessages,  

getRoutingInfo, hasMessage, isDeliveredMessage, isIncomingMessage, messageAborted,  

putToIncomingBuffer, removeFromIncomingBuffer, removeFromMessages, sendMessage,  

sortByQueueMode, toString
```

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Detail

EpidemicRouter

```
public EpidemicRouter(Settings s)
```

Constructor. Creates a new message router based on the settings in the given Settings object.

Parameters:

`s` - The settings object

EpidemicRouter

```
protected EpidemicRouter(EpidemicRouter r)
```

Copy constructor.

Parameters:

`r` - The router prototype where setting values are copied from

Method Detail

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

replicate

```
public EpidemicRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement

Class EveningActivityControlSystem

```
java.lang.Object
└ movement.EveningActivityControlSystem
```

```
public class EveningActivityControlSystem
extends java.lang.Object
```

This class controls the group mobility of the people meeting their friends in the evening

Method Summary

	void addEveningActivityNode (EveningActivityMovement eveningMovement)	Register a evening activity node with the system
static EveningActivityControlSystem	getEveningActivityControlSystem (int id)	Returns a reference to a EveningActivityControlSystem with ID provided as parameter.
EveningTrip	getEveningInstructions (int eveningActivityNodeID)	This method gets the instruction for a node, i.e.
Coord	getMeetingSpotForID (int id)	Get the meeting spot for the node
static void	reset ()	
	setMeetingSpots (java.util.List< Coord > meetingSpots)	Sets the meeting locations the nodes can choose among
	setRandomNumberGenerator (java.util.Random rand)	Sets the random number generator to be used

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Method Detail

reset

```
public static void reset()
```

addEveningActivityNode

```
public void addEveningActivityNode(EveningActivityMovement eveningMovement)
```

Register a evening activity node with the system

Parameters:

eveningMovement - activity movement

setMeetingSpots

```
public void setMeetingSpots(java.util.List<Coord> meetingSpots)
```

Sets the meeting locations the nodes can choose among

Parameters:

meetingSpots -

getEveningInstructions

```
public EveningTrip getEveningInstructions(int eveningActivityNodeID)
```

This method gets the instruction for a node, i.e. When/where and with whom to go.

Parameters:

eveningActivityNodeID - unique ID of the node

Returns:

Instructions object

getMeetingSpotForID

```
public Coord getMeetingSpotForID(int id)
```

Get the meeting spot for the node

Parameters:

id -

Returns:

Coordinates of the spot

setRandomNumberGenerator

```
public void setRandomNumberGenerator(java.util.Random rand)
```

Sets the random number generator to be used

Parameters:

rand -

getEveningActivityControlSystem

```
public static EveningActivityControlSystem getEveningActivityControlSystem(int id)
```

Returns a reference to a EveningActivityControlSystem with ID provided as parameter. If a system does not already exist with the requested ID, a new one is created.

Parameters:

id - unique ID of the EveningActivityControlSystem

Returns:

The EveningActivityControlSystem with the provided ID

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | CONSTR | [METHOD](#)

movement

Class EveningActivityMovement

```
java.lang.Object
  └── movement.MovementModel
    └── movement.MapBasedMovement
      └── movement.EveningActivityMovement
```

All Implemented Interfaces:

[SwitchableMovement](#)

```
public class EveningActivityMovement
extends MapBasedMovement
implements SwitchableMovement
```

A Class to model movement when people are out shopping or doing other activities with friends. If the node happens to be at some other location than the place where the shopping starts (where it meets its friends), it first travels to the destination along the shortest path.

Field Summary

static java.lang.String	EVENING_ACTIVITY_CONTROL_SYSTEM_NR_SETTING
static java.lang.String	MAX_GROUP_SIZE_SETTING
static java.lang.String	MAX_WAIT_TIME_SETTING
static java.lang.String	MEETING_SPOTS_FILE_SETTING
static java.lang.String	MIN_GROUP_SIZE_SETTING
static java.lang.String	MIN_WAIT_TIME_SETTING
static java.lang.String	NR_OF_MEETING_SPOTS_SETTING

Fields inherited from class movement.MapBasedMovement

[backAllowed](#), [FILE_S](#), [lastMapNode](#), [MAP_BASE_MOVEMENT_NS](#), [MAP_SELECT_S](#), [maxPathLength](#), [minPathLength](#), [NROF_FILES_S](#)

Fields inherited from class movement.MovementModel

[comBus](#), [DEF_SPEEDS](#), [DEF_WAIT_TIMES](#), [maxSpeed](#), [maxWaitTime](#), [minSpeed](#), [minWaitTime](#), [MOVEMENT_MODEL_NS](#), [rng](#), [RNG_SEED](#), [SPEED](#), [WAIT_TIME](#), [WORLD_SIZE](#)

Constructor Summary

([EveningActivityMovement](#) proto)

[EveningActivityMovement](#)

Creates a new instance of EveningActivityMovement from a prototype

[EveningActivityMovement \(Settings settings\)](#)

Creates a new instance of EveningActivityMovement

Method Summary

protected double	generateWaitTime() Generates and returns a suitable waiting time at the end of a path.
int	getID()
Coord	getInitialLocation() Returns a (random) coordinate that is between two adjacent MapNodes
Coord	getLastLocation() Get the last location the getPath() of this movement model has returned
int	getMaxGroupSize()
int	getMinGroupSize()
Path	getPath() Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
Coord	getShoppingLocation()
Coord	getShoppingLocationAndGetReady() Sets the node ready to start a shopping trip.
boolean	isReady() Checks if the movement model is finished doing its task and it's time to switch to the next movement model.
boolean	isReadyToShop() Checks if a node is at the correct place where the shopping begins
MapBasedMovement	replicate() Creates a replicate of the movement model.
static void	reset()
void	setLocation(Coord lastWaypoint) Tell the movement model what its current location is
void	setMaxGroupSize(int maxGroupSize)
void	setMinGroupSize(int minGroupSize)

Methods inherited from class movement.[MapBasedMovement](#)

[**getMap**](#), [**getOkMapNodeType**](#), [**selectRandomOkNode**](#)

Methods inherited from class movement.[MovementModel](#)

[**generateSpeed**](#), [**getComBus**](#), [**getMaxX**](#), [**getMaxY**](#), [**isActive**](#), [**nextPathAvailable**](#), [**setComBus**](#), [**toString**](#)

Methods inherited from class java.lang.Object`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`**Field Detail****NR_OF_MEETING_SPOTS_SETTING**`public static final java.lang.String NR_OF_MEETING_SPOTS_SETTING`**See Also:**[Constant Field Values](#)**EVENING_ACTIVITY_CONTROL_SYSTEM_NR_SETTING**`public static final java.lang.String EVENING_ACTIVITY_CONTROL_SYSTEM_NR_SETTING`**See Also:**[Constant Field Values](#)**MEETING_SPOTS_FILE_SETTING**`public static final java.lang.String MEETING_SPOTS_FILE_SETTING`**See Also:**[Constant Field Values](#)**MIN_GROUP_SIZE_SETTING**`public static final java.lang.String MIN_GROUP_SIZE_SETTING`**See Also:**[Constant Field Values](#)**MAX_GROUP_SIZE_SETTING**`public static final java.lang.String MAX_GROUP_SIZE_SETTING`**See Also:**[Constant Field Values](#)**MIN_WAIT_TIME_SETTING**`public static final java.lang.String MIN_WAIT_TIME_SETTING`**See Also:**[Constant Field Values](#)**MAX_WAIT_TIME_SETTING**`public static final java.lang.String MAX_WAIT_TIME_SETTING`

See Also:[Constant Field Values](#)

Constructor Detail

EveningActivityMovement

```
public EveningActivityMovement(Settings settings)
```

Creates a new instance of EveningActivityMovement

Parameters:

settings -

EveningActivityMovement

```
public EveningActivityMovement(EveningActivityMovement proto)
```

Creates a new instance of EveningActivityMovement from a prototype

Parameters:

proto -

Method Detail

getID

```
public int getID()
```

Returns:

Unique ID of the shopper

getInitialLocation

```
public Coord getInitialLocation()
```

Description copied from class: [MapBasedMovement](#)

Returns a (random) coordinate that is between two adjacent MapNodes

Overrides:

[getInitialLocation](#) in class [MapBasedMovement](#)

Returns:

The initial coordinates for a node

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Overrides:

[getPath](#) in class [MapBasedMovement](#)

Returns:

A new path or null

generateWaitTime

```
protected double generateWaitTime()
```

Description copied from class: [MovementModel](#)

Generates and returns a suitable waiting time at the end of a path. (i.e. random variable whose value is between min and max of the [MovementModel.WAIT_TIME](#) setting).

Overrides:

[generateWaitTime](#) in class [MovementModel](#)

Returns:

The time as a double

replicate

```
public MapBasedMovement replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Overrides:

[replicate](#) in class [MapBasedMovement](#)

Returns:

A new movement model with the same settings as this model

getLastLocation

```
public Coord getLastLocation()
```

Description copied from interface: [SwitchableMovement](#)

Get the last location the getPath() of this movement model has returned

Specified by:

[getLastLocation](#) in interface [SwitchableMovement](#)

Overrides:

[getLastLocation](#) in class [MapBasedMovement](#)

Returns:

the last location

See Also:

[SwitchableMovement](#)

isReady

```
public boolean isReady()
```

Description copied from interface: [SwitchableMovement](#)

Checks if the movement model is finished doing its task and it's time to switch to the next movement model. The method should be called between getPath() calls.

Specified by:

[isReady](#) in interface [SwitchableMovement](#)

Overrides:

[isReady](#) in class [MapBasedMovement](#)

Returns:

true if ready

See Also:

[SwitchableMovement](#)

setLocation

```
public void setLocation(Coord lastWaypoint)
```

Description copied from interface: [SwitchableMovement](#)

Tell the movement model what its current location is

Specified by:

[setLocation](#) in interface [SwitchableMovement](#)

Overrides:

[setLocation](#) in class [MapBasedMovement](#)

See Also:

[SwitchableMovement](#)

getShoppingLocationAndGetReady

```
public Coord getShoppingLocationAndGetReady()
```

Sets the node ready to start a shopping trip.

Returns:

The coordinate of the place where the shopping trip starts

getShoppingLocation

```
public Coord getShoppingLocation()
```

isReadyToShop

```
public boolean isReadyToShop()
```

Checks if a node is at the correct place where the shopping begins

Returns:

true if node is ready and waiting for the rest of the group to arrive

reset

```
public static void reset()
```

getMinGroupSize

```
public int getMinGroupSize()
```

setMinGroupSize

```
public void setMinGroupSize(int minGroupSize)
```

getMaxGroupSize

```
public int getMaxGroupSize()
```

setMaxGroupSize

```
public void setMaxGroupSize(int maxGroupSize)
```

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement

Class EveningTrip

```
java.lang.Object
└ movement.EveningTrip
```

```
public class EveningTrip
extends java.lang.Object
```

A class to encapsulate information about a shopping trip 1. Where the trip begins 2. Where it ends 3. The path 4. All nodes in the group

Constructor Summary

[EveningTrip\(int nrOfeveningActivityNodes, Coord location\)](#)

Create a new instance of a EveningTrip

Method Summary

boolean	addNode(EveningActivityMovement eveningActivityNode) Add an evening activity node to the group
boolean	allMembersPresent() Checks if all members of the group have found their way to the meeting point
Coord	getDestination()
Coord	getLocation()
Path	getPath()
double	getWaitTimeAtEnd()
boolean	isFull()
void	setDestination(Coord destination) Sets the destination square of the trip.
void	setPath(Path path) Sets the shopping path for the group
void	setWaitTimeAtEnd(double waitTimeAtEnd)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

EveningTrip

```
public EveningTrip(int nrOfeveningActivityNodes,
                   Coord location)
```

Create a new instance of a EveningTrip

Parameters:

nrOfeveningActivityNodes - The number of shoppers in the group
 location - Where the trip starts

Method Detail

addNode

```
public boolean addNode(EveningActivityMovement eveningActivityNode)
```

Add an evening activity node to the group

Parameters:

eveningActivityNode -

Returns:

true if there was room in the group

setPath

```
public void setPath(Path path)
```

Sets the shopping path for the group

Parameters:

path -

getPath

```
public Path getPath()
```

Returns:

The shopping trip path

getLocation

```
public Coord getLocation()
```

Returns:

The location where the shopping trip starts

isFull

```
public boolean isFull()
```

Returns:

true if the group is full

allMembersPresent

```
public boolean allMembersPresent()
```

Checks if all members of the group have found their way to the meeting point

Returns:

true if all nodes are there

getDestination

```
public Coord getDestination()
```

Returns:

The destination square of the shopping trip

setDestination

```
public void setDestination(Coord destination)
```

Sets the destination square of the trip. MUST be the same as the last node in the path

Parameters:

destination -

getWaitTimeAtEnd

```
public double getWaitTimeAtEnd()
```

setWaitTimeAtEnd

```
public void setWaitTimeAtEnd(double waitTimeAtEnd)
```

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui

Class EventLogControl

```
java.lang.Object
  ↘gui.EventLogControl
```

```
public class EventLogControl
extends java.lang.Object
```

Class encapsulates the references to the controls one can add to the EventLogControlPanel

Constructor Summary

[EventLogControl](#)(javax.swing.JCheckBox show, javax.swing.JCheckBox pause)

Constructor.

Method Summary

boolean	pauseOnEvent() Returns true if this event type should cause pause
void	setPauseOnEvent(boolean pause) Sets ought this event type cause pause (return true for pauseOnEvent())
void	 setShowEvent(boolean show) Sets ought this event type should be shown (return true for showEvent())
boolean	showEvent() Returns true if this event type should be shown

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

EventLogControl

```
public EventLogControl(javax.swing.JCheckBox show,
                      javax.swing.JCheckBox pause)
```

Constructor.

Parameters:

show - The checkbox that controls showing this type of event
pause - The checkbox that controls pausing on this type of event

Method Detail

showEvent

```
public boolean showEvent()
```

Returns true if this event type should be shown

Returns:

true if this event type should be shown

pauseOnEvent

```
public boolean pauseOnEvent()
```

Returns true if this event type should cause pause

Returns:

true if this event type should cause pause

setShowEvent

```
public void setShowEvent(boolean show)
```

Sets ought this event type should be shown (return true for [showEvent\(\)](#))

Parameters:

show - If true, events are set to be shown

setPauseOnEvent

```
public void setPauseOnEvent(boolean pause)
```

Sets ought this event type cause pause (return true for [pauseOnEvent\(\)](#))

Parameters:

pause - If true, events cause pause

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui

Class EventLogControlPanel

```
java.lang.Object
  ↘ java.awt.Component
    ↘ java.awt.Container
      ↘ javax.swing.JComponent
        ↘ javax.swing.JPanel
          ↘ gui.EventLogControlPanel
```

All Implemented Interfaces:

`java.awt.event.ActionListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible`

```
public class EventLogControlPanel
extends javax.swing.JPanel
implements java.awt.event.ActionListener
```

Control panel for event log

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JPanel

`javax.swing.JPanel.AccessibleJPanel`

Nested classes/interfaces inherited from class javax.swing.JComponent

`javax.swing.JComponent.AccessibleJComponent`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

Fields inherited from class javax.swing.JComponent

`accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW`

Fields inherited from class java.awt.Component

`BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT`

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary[EventLogControlPanel\(\)](#)

Constructor.

Method Summary

void	actionPerformed (java.awt.event.ActionEvent e)
EventLogControl	addControl (java.lang.String name) Adds a new filter&pause control with initially "show" checked but "pause" unchecked
EventLogControl	addControl (java.lang.String name, boolean showOn, boolean pauseOn) Adds a new filter&pause control
void	addHeading (java.lang.String name) Adds a new heading in the control panel.

Methods inherited from class javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

Methods inherited from class javax.swing.JComponent

```

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains,
createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange,
fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY,
getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder,
getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu,
getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics,
getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier,
getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize,
getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes,
getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor,
getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect,
getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus,
isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile,
isRequestFocusEnabled, isValidateRoot, paint, paintBorder, paintChildren, paintComponent,
paintImmediately, paintImmediately, print, printAll, printBorder, printChildren,
printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent,
processMouseEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction,
registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener,
repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow,
requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible,
setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder,
setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered,
setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap,
setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque,
setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI,
setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update

```

Methods inherited from class java.awt.Container

```

add, add, add, add, addContainerListener, addImpl, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent,
getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder,
getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout,
getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent,
processEvent, remove, remove, removeAll, removeContainerListener, setComponentZOrder,
setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout,
transferFocusBackward, transferFocusDownCycle, validate, validateTree

```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents,
dispatchEvent, enable, enableEvents, enableInputMethods, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground,
getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners,
getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests,
getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent,
getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit,
getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet,
isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable,
isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list,
location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move,
nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent,
processInputMethodEvent, processMouseWheelEvent, remove, removeComponentListener,
removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale,
 setLocation, setLocation, setName, setSize, setSize, show, show, size, toString,
transferFocus, transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail

EventLogControlPanel

```
public EventLogControlPanel()
```

Constructor. Creates a new control panel.

Method Detail

addControl

```
public EventLogControl addControl(java.lang.String name,
                                boolean showOn,
                                boolean pauseOn)
```

Adds a new filter&pause control

Parameters:

- name - Name of the control
- showOn - Is "show" initially selected
- pauseOn - Is "pause" initially selected

Returns:

Event log control object that can be queried for status

addControl

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

Adds a new filter&pause control with initially "show" checked but "pause" unchecked

Parameters:

name - Name of the control

Returns:

Event log control object that can be queried for status

See Also:

[`addControl\(String name, boolean showOn, boolean pauseOn\)`](#)

addHeading

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

Adds a new heading in the control panel. Subsequent addControl controls will be under this heading

Parameters:

name - The heading text

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

Specified by:

`actionPerformed` in interface `java.awt.event.ActionListener`

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui

Class EventLogPanel

```
java.lang.Object
  ↘ java.awt.Component
    ↘ java.awt.Container
      ↘ javax.swing.JComponent
        ↘ javax.swing.JPanel
          ↘ gui.EventLogPanel
```

All Implemented Interfaces:

[ConnectionListener](#), [MessageListener](#), java.awt.event.ActionListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible

```
public class EventLogPanel
extends javax.swing.JPanel
implements ConnectionListener, MessageListener, java.awt.event.ActionListener
```

Event log panel where log entries are displayed.

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JPanel

[javax.swing.JPanel.AccessibleJPanel](#)

Nested classes/interfaces inherited from class javax.swing.JComponent

[javax.swing.JComponent.AccessibleJComponent](#)

Nested classes/interfaces inherited from class java.awt.Container

[java.awt.Container.AccessibleAWTContainer](#)

Nested classes/interfaces inherited from class java.awt.Component

[java.awt.Component.AccessibleAWTComponent](#), [java.awt.Component.BaselineResizeBehavior](#),
[java.awt.Component.BltBufferStrategy](#), [java.awt.Component.FlipBufferStrategy](#)

Field Summary

static int	LOG_UP_INTERVAL
------------	---------------------------------

How often the log is updated (milliseconds)

Fields inherited from class javax.swing.JComponent

[accessibleContext](#), [listenerList](#), [TOOL_TIP_TEXT_KEY](#), [ui](#), [UNDEFINED_CONDITION](#),
[WHEN_ANCESTOR_OF_FOCUSED_COMPONENT](#), [WHEN_FOCUSED](#), [WHEN_IN_FOCUSED_WINDOW](#)

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary[EventLogPanel](#)([DTNSimGUI](#) gui)

Creates a new log panel

Method Summary

void	actionPerformed (java.awt.event.ActionEvent e) Action listener for log entry (host & message) buttons
EventLogControlPanel	getControls () Returns the control panel that this log uses
void	hostsConnected (DTNHost host1, DTNHost host2) Method is called when two hosts are connected.
void	hostsDisconnected (DTNHost host1, DTNHost host2) Method is called when connection between hosts is disconnected.
void	messageDeleted (Message m, DTNHost where, boolean dropped) Method is called when a message is deleted
void	messageTransferAborted (Message m, DTNHost from, DTNHost to) Method is called when a message's transfer was aborted before it finished
void	messageTransferred (Message m, DTNHost from, DTNHost to, boolean firstDelivery) Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted (Message m, DTNHost from, DTNHost to) Method is called when a message's transfer is started
void	newMessage (Message m) Method is called when a new message is created
java.lang.String	toString ()

Methods inherited from class javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder, getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile, isRequestFocusEnabled, isValidateRoot, paint, paintBorder, paintChildren, paintComponent, paintImmediately, paintImmediately, print, printAll, printBorder, printChildren,

```
printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent,
processMouseEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction,
registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener,
repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow,
requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible,
setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder,
setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered,
setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap,
setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque,
setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI,
setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update
```

Methods inherited from class java.awt.Container

```
add, add, add, add, add, addContainerListener, addImpl, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponent,
getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder,
getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout,
getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent,
processEvent, remove, remove, removeAll, removeContainerListener, setComponentZOrder,
setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout,
transferFocusBackward, transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents,
dispatchEvent, enable, enableEvents, enableInputMethods, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground,
getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners,
getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests,
getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent,
getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit,
getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet,
isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable,
isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list,
location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move,
nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent,
processInputMethodEvent, processMouseWheelEvent, remove, removeComponentListener,
removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale,
setLocation, setLocation, setName, setSize, show, show, size, transferFocus,
transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Field Detail

LOG_UP_INTERVAL

```
public static final int LOG_UP_INTERVAL
```

How often the log is updated (milliseconds)

See Also:

[Constant Field Values](#)

Constructor Detail

EventLogPanel

```
public EventLogPanel(DTNSimGUI gui)
```

Creates a new log panel

Parameters:

gui - The where this log belongs to (for callbacks)

Method Detail

getControls

```
public EventLogControlPanel getControls()
```

Returns the control panel that this log uses

Returns:

The control panel

hostsConnected

```
public void hostsConnected(DTNHost host1,
                           DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the connection

host2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost host1,
                             DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

messageDeleted

```
public void messageDeleted(Message m,
                           DTNHost where,
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

m - The message that was deleted
 where - The host where the message was deleted
 dropped - True if the message was dropped, false if removed

messageTransferred

```
public void messageTransferred(Message m,
                               DTNHost from,
                               DTNHost to,
                               boolean firstDelivery)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

m - The message that was transferred
 from - Node where the message was transferred from
 to - Node where the message was transferred to
 firstDelivery - Was the target node final destination of the message and received this message for the first time.

newMessage

```
public void newMessage(Message m)
```

Description copied from interface: [MessageListener](#)

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

m - Message that was created

messageTransferAborted

```
public void messageTransferAborted(Message m,
                                   DTNHost from,
                                   DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

m - The message that was being transferred
 from - Node where the message was being transferred from
 to - Node where the message was being transferred to

messageTransferStarted

```
public void messageTransferStarted(Message m,
                                  DTNHost from,
                                  DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

m - The message that is going to be transferred
 from - Node where the message is transferred from
 to - Node where the message is transferred to

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

Action listener for log entry (host & message) buttons

Specified by:

[actionPerformed](#) in interface [java.awt.event.ActionListener](#)

toString

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

Overrides:

[toString](#) in class [java.awt.Component](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class EventLogReport

```
java.lang.Object
  └── report.Report
      └── report.EventLogReport
```

All Implemented Interfaces:

[ConnectionListener](#), [MessageListener](#)

```
public class EventLogReport
extends Report
implements ConnectionListener, MessageListener
```

Report that creates same output as the GUI's event log panel but formatted like [StandardEventsReader](#) input. Message relying event has extra one-letter identifier to tell whether that message was delivered to final destination, delivered there again, or just normally relayed (see the public constants).

Field Summary

static java.lang.String	MESSAGE_TRANS_DELIVERED Extra info for message relayed event ("delivered"): "D"
static java.lang.String	MESSAGE_TRANS_DELIVERED AGAIN Extra info for message relayed event ("delivered again"): "A"
static java.lang.String	MESSAGE_TRANS_RELAYED Extra info for message relayed event ("relayed"): "R"

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[EventLogReport\(\)](#)

Method Summary

void	hostsConnected(DTNHost host1, DTNHost host2) Method is called when two hosts are connected.
void	hostsDisconnected(DTNHost host1, DTNHost host2) Method is called when connection between hosts is disconnected.
void	messageDeleted(Message m, DTNHost where, boolean dropped) Method is called when a message is deleted
void	messageTransferAborted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer was aborted before it finished
void	

	<code>messageTransferred(Message m, DTNHost from, DTNHost to, boolean firstDelivery)</code> Method is called when a message is successfully transferred from a node to another.
void	<code>messageTransferStarted(Message m, DTNHost from, DTNHost to)</code> Method is called when a message's transfer is started
void	<code>newMessage(Message m)</code> Method is called when a new message is created

Methods inherited from class report.Report

[`addWarmupID`](#), [`done`](#), [`format`](#), [`getAverage`](#), [`getIntAverage`](#), [`getIntMedian`](#), [`getMedian`](#), [`getScenarioName`](#), [`getSettings`](#), [`getSimTime`](#), [`getVariance`](#), [`init`](#), [`isWarmup`](#), [`isWarmupID`](#), [`newEvent`](#), [`removeWarmupID`](#), [`setPrefix`](#), [`write`](#)

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Field Detail

MESSAGE_TRANS_RELAYED

`public static final java.lang.String MESSAGE_TRANS_RELAYED`

Extra info for message relayed event ("relayed"): "R"

See Also:

[Constant Field Values](#)

MESSAGE_TRANS_DELIVERED

`public static final java.lang.String MESSAGE_TRANS_DELIVERED`

Extra info for message relayed event ("delivered"): "D"

See Also:

[Constant Field Values](#)

MESSAGE_TRANS_DELIVERED AGAIN

`public static final java.lang.String MESSAGE_TRANS_DELIVERED AGAIN`

Extra info for message relayed event ("delivered again"): "A"

See Also:

[Constant Field Values](#)

Constructor Detail

EventLogReport

`public EventLogReport()`

Method Detail

hostsConnected

```
public void hostsConnected(DTNHost host1,
                          DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the connection
 host2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost host1,
                             DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the disconnection
 host2 - Host at the other end of the connection

messageDeleted

```
public void messageDeleted(Message m,
                           DTNHost where,
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

m - The message that was deleted
 where - The host where the message was deleted
 dropped - True if the message was dropped, false if removed

messageTransferred

```
public void messageTransferred(Message m,
                               DTNHost from,
                               DTNHost to,
                               boolean firstDelivery)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

- m - The message that was transferred
 - from - Node where the message was transferred from
 - to - Node where the message was transferred to
 - firstDelivery - Was the target node final destination of the message and received this message for the first time.
-

newMessage

```
public void newMessage(Message m)
```

Description copied from interface: [MessageListener](#)

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

- m - Message that was created
-

messageTransferAborted

```
public void messageTransferAborted(Message m,  
                                 DTNHost from,  
                                 DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

- m - The message that was being transferred
 - from - Node where the message was being transferred from
 - to - Node where the message was being transferred to
-

messageTransferStarted

```
public void messageTransferStarted(Message m,  
                                 DTNHost from,  
                                 DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

- m - The message that is going to be transferred
 - from - Node where the message is transferred from
 - to - Node where the message is transferred to
-

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Interface EventQueue

All Known Implementing Classes:

[DTN2Events](#), [ExternalEventsQueue](#), [MessageBurstGenerator](#), [MessageEventGenerator](#),
[OneFromEachMessageGenerator](#), [OneToEachMessageGenerator](#), [ScheduledUpdatesQueue](#)

```
public interface EventQueue
```

Interface for event queues. Any class that is not a movement model or a routing module but wishes to provide events for the simulation (like creating messages) must implement this interface and register itself to the simulator. See the [EventQueueHandler](#) class for configuration instructions.

Method Summary

ExternalEvent double	nextEvent() nextEventsTime()
	Returns the next event in the queue or ExternalEvent with time of double.MAX_VALUE if there are no events left. Returns next event's time or Double.MAX_VALUE if there are no events left in the queue.

Method Detail

nextEvent

```
ExternalEvent nextEvent()
```

Returns the next event in the queue or ExternalEvent with time of double.MAX_VALUE if there are no events left.

Returns:

The next event

nextEventsTime

```
double nextEventsTime()
```

Returns next event's time or Double.MAX_VALUE if there are no events left in the queue.

Returns:

Next event's time

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class EventQueueHandler

```
java.lang.Object
└ input.EventQueueHandler
```

```
public class EventQueueHandler
extends java.lang.Object
```

Handler for managing event queues. Supports two different type of event queues: external event queues and event generator classes. For external event queues, the events are defined in external data file(s) (see e.g. `input.StandarEventsReader`). Event generator classes define events dynamically. Both type of event queues must implement the `input.EventQueue` interface.

The total number of event queues to load is defined with variable `NROF_SETTING`, e.g.

```
Events.nrof = 3
```

Separate event queues are configured with syntax `EventsN.variable = value` e.g.:

```
Events1.filePath = ee/messages.txt
```

or

```
Events2.class = RandomMessageGenerator
```

External event files are used when the variable `PATH_SETTING` is used to define the path to the event file and event generator class is loaded when the name of the class is defined with `CLASS_SETTING`.

Field Summary

<code>static java.lang.String</code>	CLASS PACKAGE name of the package where event generator classes are looked from
<code>static java.lang.String</code>	CLASS_SETTING name of the events class (for class based events) -setting id ("class")
<code>static java.lang.String</code>	NROF_SETTING number of event queues -setting id ("nrof")
<code>static java.lang.String</code>	PATH_SETTING path of external events file -setting id ("filePath")
<code>static java.lang.String</code>	PRELOAD_SETTING number of events to preload from file -setting id ("nrofPreload")
<code>static java.lang.String</code>	SETTINGS_NAMESPACE Event queue settings main namespace ("Events")

Constructor Summary

[`EventQueueHandler\(\)`](#)

Creates a new `EventQueueHandler` which can be queried for event queues.

Method Summary

`java.util.List<EventQueue>`

[`getEventQueues\(\)`](#)

Returns all the loaded event queues

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

SETTINGS_NAMESPACE

```
public static final java.lang.String SETTINGS_NAMESPACE
```

Event queue settings main namespace ("Events")

See Also:

[Constant Field Values](#)

NROF_SETTING

```
public static final java.lang.String NROF_SETTING
```

number of event queues -setting id ("nrof")

See Also:

[Constant Field Values](#)

CLASS_SETTING

```
public static final java.lang.String CLASS_SETTING
```

name of the events class (for class based events) -setting id ("class")

See Also:

[Constant Field Values](#)

CLASS_PACKAGE

```
public static final java.lang.String CLASS_PACKAGE
```

name of the package where event generator classes are looked from

See Also:

[Constant Field Values](#)

PRELOAD_SETTING

```
public static final java.lang.String PRELOAD_SETTING
```

number of events to preload from file -setting id ("nrofPreload")

See Also:

[Constant Field Values](#)

PATH_SETTING

```
public static final java.lang.String PATH_SETTING
```

path of external events file -setting id ("filePath")

See Also:

[Constant Field Values](#)

Constructor Detail

EventQueueHandler

```
public EventQueueHandler()
```

Creates a new EventQueueHandler which can be queried for event queues.

Method Detail

getEventQueues

```
public java.util.List<EventQueue> getEventQueues()
```

Returns all the loaded event queues

Returns:

all the loaded event queues

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement

Class ExtendedMovementModel

```
java.lang.Object
└ movement.MovementModel
    └ movement.ExtendedMovementModel
```

Direct Known Subclasses:[WorkingDayMovement](#)

```
public abstract class ExtendedMovementModel
extends MovementModel
```

Classes derived from this can make use of other movement models that implement the SwitchableMovement interface.

Field Summary

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

[ExtendedMovementModel\(\)](#)

Creates a new ExtendedMovementModel

[ExtendedMovementModel\(ExtendedMovementModel mm\)](#)

Creates a new ExtendedMovementModel from a prototype

[ExtendedMovementModel\(Settings settings\)](#)

Creates a new ExtendedMovementModel

Method Summary

protected double	generateWaitTime() Generates and returns a suitable waiting time at the end of a path.
SwitchableMovement	getCurrentMovementModel()
Path	getPath() Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
abstract boolean	newOrders() Method is called between each getPath() request when the current MM is ready (isReady() method returns true).
void	setCurrentMovementModel(SwitchableMovement mm) Sets the current movement model to be used the next time getPath() is called

Methods inherited from class movement.MovementModel

```
generateSpeed, getComBus, getInitialLocation, getMaxX, getMaxY, isActive, nextPathAvailable,  
replicate, reset, setComBus, toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail**ExtendedMovementModel**

```
public ExtendedMovementModel()
```

Creates a new ExtendedMovementModel

ExtendedMovementModel

```
public ExtendedMovementModel(Settings settings)
```

Creates a new ExtendedMovementModel

Parameters:

settings -

ExtendedMovementModel

```
public ExtendedMovementModel(ExtendedMovementModel mm)
```

Creates a new ExtendedMovementModel from a prototype

Parameters:

mm -

Method Detail**setCurrentMovementModel**

```
public void setCurrentMovementModel(SwitchableMovement mm)
```

Sets the current movement model to be used the next time getPath() is called

Parameters:

mm - Next movement model

getCurrentMovementModel

```
public SwitchableMovement getCurrentMovementModel()
```

Returns:

The movement model currently in use

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Specified by:

[getPath](#) in class [MovementModel](#)

Returns:

A new path or null

generateWaitTime

```
protected double generateWaitTime()
```

Description copied from class: [MovementModel](#)

Generates and returns a suitable waiting time at the end of a path. (i.e. random variable whose value is between min and max of the [MovementModel.WAIT_TIME](#) setting).

Overrides:

[generateWaitTime](#) in class [MovementModel](#)

Returns:

The time as a double

newOrders

```
public abstract boolean newOrders()
```

Method is called between each getPath() request when the current MM is ready (isReady() method returns true). Subclasses should implement all changes of state that need to be made here, for example switching mobility model, etc.

Returns:

true if success

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

input

Class ExternalEvent

```
java.lang.Object
└─ input.ExternalEvent
```

All Implemented Interfaces:

[java.io.Serializable](#), [java.lang.Comparable<ExternalEvent>](#)

Direct Known Subclasses:

[ConnectionEvent](#), [MessageEvent](#)

```
public class ExternalEvent
extends java.lang.Object
implements java.lang.Comparable<ExternalEvent>, java.io.Serializable
```

Super class for all external events. All new classes of external events must extend this class. This can also be used as a dummy event if only an update request (and no further actions) to all hosts is needed.

See Also:

[Serialized Form](#)

Field Summary

protected double	time Time of the event (simulated seconds)
------------------	---

Constructor Summary

[ExternalEvent](#)(double time)

Method Summary

int	compareTo(ExternalEvent other) Compares two external events by their time.
double	getTime() Returns the time when this event should happen.
void	processEvent(World world) Processes the external event.
java.lang.String	toString() Returns a String representation of the event

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

time

```
protected double time
```

Time of the event (simulated seconds)

Constructor Detail

ExternalEvent

```
public ExternalEvent(double time)
```

Method Detail

processEvent

```
public void processEvent(World world)
```

Processes the external event.

Parameters:

`world` - World where the actors of the event are

getTime

```
public double getTime()
```

Returns the time when this event should happen.

Returns:

Event's time

compareTo

```
public int compareTo(ExternalEvent other)
```

Compares two external events by their time.

Specified by:

`compareTo` in interface `java.lang.Comparable<ExternalEvent>`

Parameters:

`other` - The other external event

Returns:

-1, zero, 1 if this event happens before, at the same time, or after the other event

toString

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

Returns a String representation of the event

Overrides:

`toString` in class `java.lang.Object`

Returns:

a String representation of the event

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class ExternalEventsQueue

```
java.lang.Object
└─ input.ExternalEventsQueue
```

All Implemented Interfaces:[EventQueue](#)

```
public class ExternalEventsQueue
extends java.lang.Object
implements EventQueue
```

Queue of external events. This class also takes care of buffering the events and preloading only a proper amount of them.

Field Summary

static int	DEFAULT_NROF_PRELOAD default number of preloaded events
static java.lang.String	PATH_SETTING path of external events file -setting id ("filePath")
static java.lang.String	PRELOAD_SETTING number of event to preload -setting id ("nrofPreload")
static java.lang.String	SETTINGS_NAMESPACE ExternalEvents namespace ("ExternalEvents")

Constructor Summary

[ExternalEventsQueue\(Settings s\)](#)Create a new Queue based on the given settings: [PRELOAD_SETTING](#) and [PATH_SETTING](#).[ExternalEventsQueue\(java.lang.String filePath, int nrofPreload\)](#)

Creates a new Queue from a file

Method Summary

int	eventsLeftInBuffer() Returns the amount of events left in the buffer at the moment (the amount can increase later if more events are read).
ExternalEvent	nextEvent() Returns the next event in the queue or ExternalEvent with time of double.MAX_VALUE if there are no events left
double	nextEventsTime() Returns next event's time or Double.MAX_VALUE if there are no events left
void	setNrofPreload(int nrof) Sets maximum number of events that are read when the next preload occurs

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

Field Detail**SETTINGS_NAMESPACE**

```
public static final java.lang.String SETTINGS_NAMESPACE
```

ExternalEvents namespace ("ExternalEvents")

See Also:

[Constant Field Values](#)

PRELOAD_SETTING

```
public static final java.lang.String PRELOAD_SETTING
```

number of event to preload -setting id ("nrofPreload")

See Also:

[Constant Field Values](#)

PATH_SETTING

```
public static final java.lang.String PATH_SETTING
```

path of external events file -setting id ("filePath")

See Also:

[Constant Field Values](#)

DEFAULT_NROF_PRELOAD

```
public static final int DEFAULT_NROF_PRELOAD
```

default number of preloaded events

See Also:

[Constant Field Values](#)

Constructor Detail**ExternalEventsQueue**

```
public ExternalEventsQueue(java.lang.String filePath,
                           int nrofPreload)
```

Creates a new Queue from a file

Parameters:

`filePath` - Path to the file where the events are read from. If file ends with extension defined in [BinaryEventsReader.BINARY_EXT](#) the file is assumed to be a binary file.
`nrofPreload` - How many events to preload

See Also:

[BinaryEventsReader.BINARY_EXT](#), [BinaryEventsReader.storeToBinaryFile\(String, List\)](#)

ExternalEventsQueue

```
public ExternalEventsQueue(Settings s)
```

Create a new Queue based on the given settings: [PRELOAD_SETTING](#) and [PATH_SETTING](#). The path setting supports value filling.

Parameters:

`s` - The settings

Method Detail

setNrofPreload

```
public void setNrofPreload(int nrof)
```

Sets maximum number of events that are read when the next preload occurs

Parameters:

`nrof` - Maximum number of events to read. If less than 1, default value ([500](#)) is used.

nextEventsTime

```
public double nextEventsTime()
```

Returns next event's time or Double.MAX_VALUE if there are no events left

Specified by:

[nextEventsTime](#) in interface [EventQueue](#)

Returns:

Next event's time

nextEvent

```
public ExternalEvent nextEvent()
```

Returns the next event in the queue or ExternalEvent with time of double.MAX_VALUE if there are no events left

Specified by:

[nextEvent](#) in interface [EventQueue](#)

Returns:

The next event

eventsLeftInBuffer

```
public int eventsLeftInBuffer()
```

Returns the amount of events left in the buffer at the moment (the amount can increase later if more events are read).

Returns:

The amount of events left or 0 there aren't any events

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Interface ExternalEventsReader

All Known Implementing Classes:

[BinaryEventsReader](#), [StandardEventsReader](#)

```
public interface ExternalEventsReader
```

Interface for external event readers.

Method Summary

<pre>void</pre>	close()	Closes the input file streams of the reader.
<pre>java.util.List<ExternalEvent></pre>	readEvents(int nrof)	Read events from the reader

Method Detail

readEvents

```
java.util.List<ExternalEvent> readEvents(int nrof)
```

Read events from the reader

Parameters:

nrof - Maximum number of events to read

Returns:

Events in a List

close

```
void close()
```

Closes the input file streams of the reader.

movement

Class ExternalMovement

```
java.lang.Object
└ movement.MovementModel
    └ movement.ExternalMovement
```

```
public class ExternalMovement
extends MovementModel
```

Movement model that uses external data of node locations.

Field Summary

static java.lang.String	EXTERNAL_MOVEMENT_NS Namespace for settings
static java.lang.String	MOVEMENT_FILE_S external locations file's path -setting id ("file")
static java.lang.String	NROF_PRELOAD_S number of preloaded intervals per preload run -setting id ("nrofPreload")

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

[ExternalMovement](#)(Settings settings)

Constructor for the prototype.

Method Summary

Coord	getInitialLocation() Returns a new initial placement for a node
int	getMaxX() Returns the largest X coordinate value this model uses
int	getMaxY() Returns the largest Y coordinate value this model uses
Path	getPath() Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
boolean	isActive() Returns true if this node is active at the moment (false if not)
double	nextPathAvailable() Returns a sim time when the next path is available.

MovementModel	<u>replicate()</u>	Creates a replicate of the movement model.
static void	<u>reset()</u>	Reset state so that next instance will have a fresh state

Methods inherited from class movement.[MovementModel](#)[generateSpeed](#), [generateWaitTime](#), [getComBus](#), [setComBus](#), [toString](#)**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

EXTERNAL_MOVEMENT_NS

public static final java.lang.String **EXTERNAL_MOVEMENT_NS**

Namespace for settings

See Also:[Constant Field Values](#)

MOVEMENT_FILE_S

public static final java.lang.String **MOVEMENT_FILE_S**

external locations file's path -setting id ("file")

See Also:[Constant Field Values](#)

NROF_PRELOAD_S

public static final java.lang.String **NROF_PRELOAD_S**

number of preloaded intervals per preload run -setting id ("nrofPreload")

See Also:[Constant Field Values](#)

Constructor Detail

ExternalMovement

public **ExternalMovement**([Settings](#) settings)

Constructor for the prototype. Run once per group.

Parameters:

settings - Where settings are read from

Method Detail

getInitialLocation

```
public Coord getInitialLocation()
```

Description copied from class: [MovementModel](#)

Returns a new initial placement for a node

Specified by:

[getInitialLocation](#) in class [MovementModel](#)

Returns:

The initial coordinates for a node

isActive

```
public boolean isActive()
```

Description copied from class: [MovementModel](#)

Returns true if this node is active at the moment (false if not)

Overrides:

[isActive](#) in class [MovementModel](#)

Returns:

true if this node is active (false if not)

nextPathAvailable

```
public double nextPathAvailable()
```

Returns a sim time when the next path is available.

Overrides:

[nextPathAvailable](#) in class [MovementModel](#)

Returns:

The sim time when node should ask the next time for a path

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Specified by:

[getPath](#) in class [MovementModel](#)

Returns:

A new path or null

getMaxX

```
public int getMaxX()
```

Description copied from class: [MovementModel](#)

Returns the largest X coordinate value this model uses

Overrides:

[getMaxX](#) in class [MovementModel](#)

Returns:

Maximum of X coordinate values

getMaxY

```
public int getMaxY()
```

Description copied from class: [MovementModel](#)

Returns the largest Y coordinate value this model uses

Overrides:

[getMaxY](#) in class [MovementModel](#)

Returns:

Maximum of Y coordinate values

replicate

```
public MovementModel replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Specified by:

[replicate](#) in class [MovementModel](#)

Returns:

A new movement model with the same settings as this model

reset

```
public static void reset()
```

Reset state so that next instance will have a fresh state

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class ExternalMovementReader

```
java.lang.Object
└ input.ExternalMovementReader
```

```
public class ExternalMovementReader
extends java.lang.Object
```

Reader for ExternalMovement movement model's time-location tuples.

First line of the file should be the offset header. Syntax of the header should be:

minTime maxTime minX maxX minY maxY minZ maxZ

Last two values (Z-axis) are ignored at the moment but can be present in the file.

Following lines' syntax should be:

time id xPos yPos

where `time` is the time when a node with `id` should be at location `(xPos, yPos)`.

All lines must be sorted by time. Sampling interval (time difference between two time instances) must be same for the whole file.

Field Summary

static java.lang.String	COMMENT_PREFIX
-------------------------	--------------------------------

Constructor Summary

[ExternalMovementReader](#)(java.lang.String inFilePath)

Constructor.

Method Summary

double	getLastTimeStamp() Returns the time stamp where the last moves read with readNextMovements() belong to.
double	getMaxTime() Returns offset maxTime
double	getMaxX() Returns offset maxX
double	getMaxY() Returns offset maxY
double	getMinTime() Returns offset minTime
double	getMinX() Returns offset minX

	double	getMinY()	Returns offset minY
java.util.List< Tuple <java.lang.String, Coord >>		readNextMovements()	Reads all new id-coordinate tuples that belong to the same time instance
	void	setNormalize(boolean normalize)	Sets normalizing of read values on/off.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

COMMENT_PREFIX

```
public static final java.lang.String COMMENT_PREFIX
```

See Also:

[Constant Field Values](#)

Constructor Detail

ExternalMovementReader

```
public ExternalMovementReader(java.lang.String inFilePath)
```

Constructor. Creates a new reader that reads the data from a file.

Parameters:

inFilePath - Path to the file where the data is read

Throws:

[SettingsError](#) - if the file wasn't found

Method Detail

setNormalize

```
public void setNormalize(boolean normalize)
```

Sets normalizing of read values on/off. If on, values returned by [readNextMovements\(\)](#) are decremented by minimum values of the offsets. Default is on (normalize).

Parameters:

normalize - If true, normalizing is on (false -> off).

readNextMovements

```
public java.util.List<Tuple<java.lang.String, Coord>> readNextMovements()
```

Reads all new id-coordinate tuples that belong to the same time instance

Returns:

A list of tuples or empty list if there were no more moves

Throws:

SettingError - if an invalid line was read

getLastTimeStamp

```
public double getLastTimeStamp()
```

Returns the time stamp where the last moves read with [readNextMovements\(\)](#) belong to.

Returns:

The time stamp

getMaxTime

```
public double getMaxTime()
```

Returns offset maxTime

Returns:

the maxTime

getMaxX

```
public double getMaxX()
```

Returns offset maxX

Returns:

the maxX

getMaxY

```
public double getMaxY()
```

Returns offset maxY

Returns:

the maxY

getMinTime

```
public double getMinTime()
```

Returns offset minTime

Returns:

the minTime

getMinX

```
public double getMinX()
```

Returns offset minX

Returns:

the minX

getMinY

```
public double getMinY()
```

Returns offset minY

Returns:

the minY

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

routing

Class FirstContactRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.FirstContactRouter
```

```
public class FirstContactRouter
extends ActiveRouter
```

First contact router which uses only a single copy of the message (or fragments) and forwards it to the first available contact.

Field Summary

Fields inherited from class routing.ActiveRouter

[DELETE_DELIVERED_S](#), [deleteDelivered](#), [RESPONSE_PREFIX](#), [sendingConnections](#), [TTL_CHECK_INTERVAL](#)

Fields inherited from class routing.MessageRouter

[B_SIZE_S](#), [DENIED_NO_SPACE](#), [DENIED_OLD](#), [DENIED_TTL](#), [DENIED_UNSPECIFIED](#), [MSG_TTL_S](#), [msgTtl](#),
[Q_MODE_FIFO](#), [Q_MODE_RANDOM](#), [RCV_OK](#), [SEND_QUEUE_MODE_S](#), [TRY_LATER_BUSY](#)

Constructor Summary

protected	FirstContactRouter(FirstContactRouter r) Copy constructor.
	FirstContactRouter(Settings s) Constructor.

Method Summary

protected int	checkReceiving(Message m) Checks if router "wants" to start receiving message (i.e.
FirstContactRouter	replicate() Creates a replicate of this router.
protected void	transferDone(Connection con) Method is called just before a transfer is finalized at ActiveRouter.update() .
void	update() Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.ActiveRouter

[addToSendingsConnections](#), [canStartTransfer](#), [changedConnection](#), [createNewMessage](#),
[dropExpiredMessages](#), [exchangeDeliverableMessages](#), [getConnections](#), [getMessagesForConnected](#),
[getOldestMessage](#), [init](#), [isSending](#), [isTransferring](#), [makeRoomForMessage](#), [makeRoomForNewMessage](#),

```
messageTransferred, receiveMessage, requestDeliverableMessages, shuffleMessages,  

startTransfer, transferAborted, tryAllMessages, tryAllMessagesToAllConnections,  

tryMessagesForConnected, tryMessagesToConnections
```

Methods inherited from class routing.MessageRouter

```
addApplication, addToMessages, compareByQueueMode, deleteMessage, getApplications,  

getBufferSize, getFreeBufferSize, getHost, getMessage, getMessageCollection, getNrofMessages,  

getRoutingInfo, hasMessage, isDeliveredMessage, isIncomingMessage, messageAborted,  

putToIncomingBuffer, removeFromIncomingBuffer, removeFromMessages, sendMessage,  

sortByQueueMode, toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail

FirstContactRouter

```
public FirstContactRouter(Settings s)
```

Constructor. Creates a new message router based on the settings in the given Settings object.

Parameters:

s - The settings object

FirstContactRouter

```
protected FirstContactRouter(FirstContactRouter r)
```

Copy constructor.

Parameters:

r - The router prototype where setting values are copied from

Method Detail

checkReceiving

```
protected int checkReceiving(Message m)
```

Description copied from class: [ActiveRouter](#)

Checks if router "wants" to start receiving message (i.e. router isn't transferring, doesn't have the message and has room for it).

Overrides:

[checkReceiving](#) in class [ActiveRouter](#)

Parameters:

m - The message to check

Returns:

A return code similar to [MessageRouter.receiveMessage\(Message, DTNHost\)](#), i.e. [MessageRouter.RCV_OK](#) if receiving seems to be OK, TRY_LATER_BUSY if router is transferring, DENIED_OLD if the router is already carrying the message or it has been delivered to this router (as final recipient), or DENIED_NO_SPACE if the message does not fit into buffer

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

transferDone

```
protected void transferDone(Connection con)
```

Description copied from class: [ActiveRouter](#)

Method is called just before a transfer is finalized at [ActiveRouter.update\(\)](#). Subclasses that are interested of the event may want to override this.

Overrides:

[transferDone](#) in class [ActiveRouter](#)

Parameters:

con - The connection whose transfer was finalized

replicate

```
public FirstContactRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui

Class GUIControls

```
java.lang.Object
  ↘ java.awt.Component
    ↘ java.awt.Container
      ↘ javax.swing.JComponent
        ↘ javax.swing.JPanel
          ↘ gui.GUIControls
```

All Implemented Interfaces:

`java.awt.event.ActionListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible, javax.swing.event.ChangeListener`

```
public class GUIControls
extends javax.swing.JPanel
implements java.awt.event.ActionListener, javax.swing.event.ChangeListener
```

GUI's control panel

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JPanel

`javax.swing.JPanel.AccessibleJPanel`

Nested classes/interfaces inherited from class javax.swing.JComponent

`javax.swing.JComponent.AccessibleJComponent`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

<code>static int</code>	FFW SPEED INDEX index of FFW speed setting
<code>static int</code>	INITIAL SPEED SELECTION index of initial update speed setting
<code>static java.lang.String[]</code>	UP SPEEDS GUI update speeds.

static double	ZOOM_MAX	Highest value for the zoom level
static double	ZOOM_MIN	Smallest value for the zoom level

Fields inherited from class javax.swing.JComponent

```
accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION,
WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW
```

Fields inherited from class java.awt.Component

```
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT
```

Fields inherited from interface java.awt.image.ImageObserver

```
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH
```

Constructor Summary

[GUIControls](#)([DTNSimGUI](#) gui, [PlayField](#) pf)

Method Summary

void	actionPerformed (java.awt.event.ActionEvent e)
void	changeZoom (int delta) Changes the zoom level
double	getUpdateInterval () Returns the selected update interval of GUI
boolean	isFFw () Is fast forward turned on
boolean	isPaused () Has user requested the simulation to be paused
void	setPaused (boolean paused) Sets simulation to pause or play.
void	setSimTime (double time) Sets the simulation time that control panel shows
void	stateChanged (javax.swing.event.ChangeEvent e)

Methods inherited from class javax.swing.JPanel

```
getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI
```

Methods inherited from class javax.swing.JComponent

```
addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains,
createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange,
fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY,
getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder,
getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu,
getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics,
getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier,
getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize,
getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes,
getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor,
```

```
getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect,
getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus,
isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile,
isRequestFocusEnabled, isValidateRoot, paint, paintBorder, paintChildren, paintComponent,
paintImmediately, paintImmediately, print, printAll, printBorder, printChildren,
printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent,
processMouseEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction,
registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener,
repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow,
requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible,
setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder,
setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered,
setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap,
setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque,
setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI,
setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update
```

Methods inherited from class java.awt.Container

```
add, add, add, add, addContainerListener, addImpl, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent,
getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder,
getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout,
getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent,
processEvent, remove, remove, removeAll, removeContainerListener, setComponentZOrder,
setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout,
transferFocusBackward, transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents,
dispatchEvent, enable, enableEvents, enableInputMethods, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground,
getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners,
getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests,
getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent,
getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit,
getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet,
isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable,
isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list,
location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move,
nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent,
processInputMethodEvent, processMouseWheelEvent, remove, removeComponentListener,
removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale,
setLocation, setLocation, setName, setSize, setSize, show, show, size, toString,
transferFocus, transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Field Detail

UP_SPEEDS

```
public static final java.lang.String[] UP_SPEEDS
```

GUI update speeds. Negative values -> how many 1/10 seconds to wait between updates. Positive values ->

show every Nth update

ZOOM_MIN

```
public static final double ZOOM_MIN
```

Smallest value for the zoom level

See Also:

[Constant Field Values](#)

ZOOM_MAX

```
public static final double ZOOM_MAX
```

Highest value for the zoom level

See Also:

[Constant Field Values](#)

INITIAL_SPEED_SELECTION

```
public static final int INITIAL_SPEED_SELECTION
```

index of initial update speed setting

See Also:

[Constant Field Values](#)

FFW_SPEED_INDEX

```
public static final int FFW_SPEED_INDEX
```

index of FFW speed setting

See Also:

[Constant Field Values](#)

Constructor Detail

GUIControls

```
public GUIControls(DTNSimGUI gui,  
                   PlayField pf)
```

Method Detail

setSimTime

```
public void setSimTime(double time)
```

Sets the simulation time that control panel shows

Parameters:

time - The time to show

setPaused

```
public void setPaused(boolean paused)
```

Sets simulation to pause or play.

Parameters:

paused - If true, simulation is put to pause

isPaused

```
public boolean isPaused()
```

Has user requested the simulation to be paused

Returns:

True if pause is requested

isFFw

```
public boolean isFFw()
```

Is fast forward turned on

Returns:

True if FFW is on, false if not

getUpdateInterval

```
public double getUpdateInterval()
```

Returns the selected update interval of GUI

Returns:

The update interval (seconds)

changeZoom

```
public void changeZoom(int delta)
```

Changes the zoom level

Parameters:

delta - How much to change the current level (can be negative or positive)

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

Specified by:

actionPerformed in interface java.awt.event.ActionListener

stateChanged

public void **stateChanged**(javax.swing.event.ChangeEvent e)

Specified by:

stateChanged in interface javax.swing.event.ChangeListener

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement

Class HomeActivityMovement

```
java.lang.Object
└ movement.MovementModel
  └ movement.MapBasedMovement
    └ movement.HomeActivityMovement
```

All Implemented Interfaces:

[SwitchableMovement](#)

```
public class HomeActivityMovement
extends MapBasedMovement
implements SwitchableMovement
```

A Class to model movement at home. If the node happens to be at some other location than its home, it first walks the shortest path home location and then stays there until morning. A node has only one home

Field Summary

static java.lang.String	HOME_LOCATIONS_FILE_SETTING
static java.lang.String	STD_FOR_TIME_DIFF_SETTING

Fields inherited from class movement.MapBasedMovement

```
backAllowed, FILE_S, lastMapNode, MAP_BASE_MOVEMENT_NS, MAP_SELECT_S, maxPathLength,
minPathLength, NROF_FILES_S
```

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

[HomeActivityMovement](#)([HomeActivityMovement](#) proto)

Creates a new instance of HomeActivityMovement from a prototype

[HomeActivityMovement](#)([Settings](#) settings)

Creates a new instance of HomeActivityMovement

Method Summary

protected double	generateWaitTime() Generates and returns a suitable waiting time at the end of a path.
Coord	getHomeLocation()
Coord	getInitialLocation()

	Returns a (random) coordinate that is between two adjacent MapNodes
Coord	getLastLocation() Get the last location the getPath() of this movement model has returned
Path	getPath() Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
boolean	isReady() Checks if the movement model is finished doing its task and it's time to switch to the next movement model.
MapBasedMovement	replicate() Creates a replicate of the movement model.
void	setLocation(Coord lastWaypoint) Tell the movement model what its current location is

Methods inherited from class movement.[MapBasedMovement](#)[getMap](#), [getOkMapNodeType](#)s, [selectRandomOkNode](#)**Methods inherited from class movement.[MovementModel](#)**[generateSpeed](#), [getComBus](#), [getMaxX](#), [getMaxY](#), [isActive](#), [nextPathAvailable](#), [reset](#), [setComBus](#), [toString](#)**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

HOME_LOCATIONS_FILE_SETTING

public static final java.lang.String **HOME_LOCATIONS_FILE_SETTING****See Also:**[Constant Field Values](#)

STD_FOR_TIME_DIFF_SETTING

public static final java.lang.String **STD_FOR_TIME_DIFF_SETTING****See Also:**[Constant Field Values](#)

Constructor Detail

HomeActivityMovement

public **HomeActivityMovement**([Settings](#) settings)

Creates a new instance of HomeActivityMovement

Parameters:

settings -

HomeActivityMovement

```
public HomeActivityMovement(HomeActivityMovement proto)
```

Creates a new instance of HomeActivityMovement from a prototype

Parameters:

proto -

Method Detail

getInitialLocation

```
public Coord getInitialLocation()
```

Description copied from class: [MapBasedMovement](#)

Returns a (random) coordinate that is between two adjacent MapNodes

Overrides:

[getInitialLocation](#) in class [MapBasedMovement](#)

Returns:

The initial coordinates for a node

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Overrides:

[getPath](#) in class [MapBasedMovement](#)

Returns:

A new path or null

generateWaitTime

```
protected double generateWaitTime()
```

Description copied from class: [MovementModel](#)

Generates and returns a suitable waiting time at the end of a path. (i.e. random variable whose value is between min and max of the [MovementModel.WAIT_TIME](#) setting).

Overrides:

[generateWaitTime](#) in class [MovementModel](#)

Returns:

The time as a double

replicate

```
public MapBasedMovement replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Overrides:

[replicate](#) in class [MapBasedMovement](#)

Returns:

A new movement model with the same settings as this model

getLastLocation

```
public Coord getLastLocation()
```

Description copied from interface: [SwitchableMovement](#)

Get the last location the getPath() of this movement model has returned

Specified by:

[getLastLocation](#) in interface [SwitchableMovement](#)

Overrides:

[getLastLocation](#) in class [MapBasedMovement](#)

Returns:

the last location

See Also:

[SwitchableMovement](#)

isReady

```
public boolean isReady()
```

Description copied from interface: [SwitchableMovement](#)

Checks if the movement model is finished doing its task and it's time to switch to the next movement model. The method should be called between getPath() calls.

Specified by:

[isReady](#) in interface [SwitchableMovement](#)

Overrides:

[isReady](#) in class [MapBasedMovement](#)

Returns:

true if ready

See Also:

[SwitchableMovement](#)

setLocation

```
public void setLocation(Coord lastWaypoint)
```

Description copied from interface: [SwitchableMovement](#)

Tell the movement model what its current location is

Specified by:

[setLocation](#) in interface [SwitchableMovement](#)

Overrides:

[setLocation](#) in class [MapBasedMovement](#)

See Also:

[SwitchableMovement](#)

getHomeLocation

public [Coord](#) [getHomeLocation\(\)](#)

Returns:

Home location of the node

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

gui

Class InfoPanel

```
java.lang.Object
  ↘ java.awt.Component
    ↘ java.awt.Container
      ↘ javax.swing.JComponent
        ↘ javax.swing.JPanel
          ↘ gui.InfoPanel
```

All Implemented Interfaces:

`java.awt.event.ActionListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible`

```
public class InfoPanel
extends javax.swing.JPanel
implements java.awt.event.ActionListener
```

Information panel that shows data of selected messages and nodes.

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JPanel

`javax.swing.JPanel.AccessibleJPanel`

Nested classes/interfaces inherited from class javax.swing.JComponent

`javax.swing.JComponent.AccessibleJComponent`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

Fields inherited from class javax.swing.JComponent

`accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW`

Fields inherited from class java.awt.Component

`BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT`

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary[InfoPanel](#) ([DTNSimGUI](#) gui)**Method Summary**void [actionPerformed](#)(java.awt.event.ActionEvent e)void [showInfo](#)([DTNHost](#) host)
Show information about a hostvoid [showInfo](#)([Message](#) message)
Show information about a message**Methods inherited from class javax.swing.JPanel**

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder, getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getSize, getToolTipLocation, getToolTipText, getTopLevelAncestor, getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile, isRequestFocusEnabled, isValidateRoot, paint, paintBorder, paintChildren, paintComponent, paintImmediately, paintImmediately, print, printAll, printBorder, printChildren, printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent, processMouseEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update

Methods inherited from class java.awt.Container

add, add, add, add, addContainerListener, addImpl, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout, getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent, processEvent, remove, remove, removeAll, removeContainerListener, setComponentZOrder, setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout, transferFocusBackward, transferFocusDownCycle, validate, validateTree

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,

```

addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents,
dispatchEvent, enable, enableEvents, enableInputMethods, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground,
getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners,
getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests,
getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent,
getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit,
getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet,
isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable,
isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isPreferredSizeSet, isShowing, isValid, keyDown, keyUp, list, list, list,
location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move,
nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent,
processInputMethodEvent, processMouseWheelEvent, remove, removeComponentListener,
removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale,
setLocation, setLocation, setName, setSize, setSize, show, show, size, toString,
transferFocus, transferFocusUpCycle

```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail

InfoPanel

```
public InfoPanel(DTNSimGUI gui)
```

Method Detail

showInfo

```
public void showInfo(DTNHost host)
```

Show information about a host

Parameters:

host - Host to show the information of

showInfo

```
public void showInfo(Message message)
```

Show information about a message

Parameters:

message - Message to show the information of

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

Specified by:

actionPerformed in interface java.awt.event.ActionListener

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class InterContactTimesReport

```
java.lang.Object
└ report.Report
  └ report.ContactTimesReport
    └ report.InterContactTimesReport
```

All Implemented Interfaces:

[ConnectionListener](#)

```
public class InterContactTimesReport
extends ContactTimesReport
```

Reports the inter-contact time (i.e., the time between the end of previous contact and the beginning of a new contact between two hosts) distribution. The syntax of the report file is the same as in [ContactTimesReport](#).

Nested Class Summary

Nested classes/interfaces inherited from class report.ContactTimesReport

[ContactTimesReport.ConnectionInfo](#)

Field Summary

Fields inherited from class report.ContactTimesReport

[connections](#), [granularity](#), [GRANULARITY](#)

Fields inherited from class report.Report

[DEF_PRECISION](#), [INTERVAL_SETTING](#), [INTERVALLED_FORMAT](#), [NAN](#), [out](#), [OUT_SUFFIX](#), [OUTPUT_SETTING](#), [PRECISION_SETTING](#), [REPORT_NS](#), [REPORTDIR_SETTING](#), [WARMUP_S](#), [warmupIDs](#), [warmupTime](#)

Constructor Summary

[InterContactTimesReport\(\)](#)

Method Summary

void [hostsConnected\(DTNHost host1, DTNHost host2\)](#)

Method is called when two hosts are connected.

void [hostsDisconnected\(DTNHost host1, DTNHost host2\)](#)

Method is called when connection between hosts is disconnected.

Methods inherited from class report.ContactTimesReport

addConnection , done , increaseTimeCount , init , removeConnection
--

Methods inherited from class report.Report

addWarmupID , format , getAverage , getIntAverage , getIntMedian , getMedian , getScenarioName , getSettings , getSimTime , getVariance , isWarmup , isWarmupID , newEvent , removeWarmupID , setPrefix , write

Methods inherited from class java.lang.Object

clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait
--

Constructor Detail

InterContactTimesReport

```
public InterContactTimesReport()
```

Method Detail

hostsConnected

```
public void hostsConnected(DTNHost host1,  

DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Overrides:

[hostsConnected](#) in class [ContactTimesReport](#)

Parameters:

host1 - Host that initiated the connection

host2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost host1,  

DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Overrides:

[hostsDisconnected](#) in class [ContactTimesReport](#)

Parameters:

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

interfaces

Class InterferenceLimitedInterface

```
java.lang.Object
└ core.NetworkInterface
    └ interfaces.InterferenceLimitedInterface
```

All Implemented Interfaces:[ModuleCommunicationListener](#)

```
public class InterferenceLimitedInterface
extends NetworkInterface
```

A simple Network Interface that provides a variable bit-rate service, where the bit-rate depends on the number of other transmitting stations within range. The current transmit speed is updated only if there are ongoing transmissions. The configured transmit speed is the maximum obtainable speed.

Field Summary

protected int	currentTransmitSpeed
protected int	numberOfTransmissions

Fields inherited from class core.NetworkInterface

```
connections, host, interfaceType, optimizer, RANGE_ID, SCAN_INTERVAL_ID, SCAN_INTERVAL_S,
SPEED_ID, TRANSMIT_RANGE_S, TRANSMIT_SPEED_S, transmitRange, transmitSpeed
```

Constructor Summary

[InterferenceLimitedInterface](#)([InterferenceLimitedInterface](#) ni)

Copy constructor

[InterferenceLimitedInterface](#)([Settings](#) s)

Method Summary

void	connect (NetworkInterface anotherInterface) Tries to connect this host to another host.
void	createConnection (NetworkInterface anotherInterface) Creates a connection to another host.
int	getTransmitSpeed () Returns the transmit speed of this network layer
boolean	isTransferring () Returns true if this interface is actually transmitting data
NetworkInterface	replicate ()

Replication function	
java.lang.String	toString() Returns a string representation of the object.
void	update() Updates the state of current connections (i.e., tears down connections that are out of range).

Methods inherited from class core.[NetworkInterface](#)

[connect](#), [destroyConnection](#), [disconnect](#), [ensurePositiveValue](#), [getAddress](#), [getConnections](#), [getHost](#), [getInterfaceType](#), [getLocation](#), [getTransmitRange](#), [isConnected](#), [isScanning](#), [isWithinRange](#), [moduleValueChanged](#), [reset](#), [setClisteners](#), [setHost](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

currentTransmitSpeed

protected int **currentTransmitSpeed**

numberOfTransmissions

protected int **numberOfTransmissions**

Constructor Detail

InterferenceLimitedInterface

public **InterferenceLimitedInterface**([Settings](#) s)

InterferenceLimitedInterface

public **InterferenceLimitedInterface**([InterferenceLimitedInterface](#) ni)

Copy constructor

Parameters:

ni - the copied network interface object

Method Detail

replicate

public [NetworkInterface](#) **replicate()**

Description copied from class: [NetworkInterface](#)

Replication function

Specified by:

[replicate](#) in class [NetworkInterface](#)

getTransmitSpeed

`public int getTransmitSpeed()`

Returns the transmit speed of this network layer

Overrides:

[getTransmitSpeed](#) in class [NetworkInterface](#)

Returns:

the transmit speed

connect

`public void connect(NetworkInterface anotherInterface)`

Tries to connect this host to another host. The other host must be active and within range of this host for the connection to succeed.

Specified by:

[connect](#) in class [NetworkInterface](#)

Parameters:

`anotherInterface` - The host to connect to

update

`public void update()`

Updates the state of current connections (i.e., tears down connections that are out of range).

Specified by:

[update](#) in class [NetworkInterface](#)

createConnection

`public void createConnection(NetworkInterface anotherInterface)`

Creates a connection to another host. This method does not do any checks on whether the other node is in range or active

Specified by:

[createConnection](#) in class [NetworkInterface](#)

Parameters:

`anotherInterface` - The interface to create the connection to

isTransferring

`public boolean isTransferring()`

Returns true if this interface is actually transmitting data

toString

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

Returns a string representation of the object.

Overrides:

[toString](#) in class [NetworkInterface](#)

Returns:

a string representation of the object.

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement

Class LinearFormation

```
java.lang.Object
└ movement.MovementModel
    └ movement.LinearFormation
```

```
public class LinearFormation
extends MovementModel
```

A stationary "movement" model where nodes do not move but are in linear formation (i.e., in a line).

Field Summary

static java.lang.String	END_LOCATION_S Per node group setting for defining the end coordinates of the line ("endLocation")
static java.lang.String	LINEAR_FORMATION_NS Name space of the settings (append to group name space)
static java.lang.String	START_LOCATION_S Per node group setting for defining the start coordinates of the line ("startLocation")

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

[LinearFormation\(LinearFormation lf\)](#)

Copy constructor.

[LinearFormation\(Settings s\)](#)

Creates a new movement model based on a Settings object's settings.

Method Summary

Coord	getInitialLocation() Returns the the location of the node in the formation
Path	getPath() Returns a single coordinate path (using the only possible coordinate)
double	nextPathAvailable() Returns Double.MAX_VALUE (no paths available)
LinearFormation	replicate() Creates a replicate of the movement model.

Methods inherited from class movement.MovementModel

```
generateSpeed, generateWaitTime, getComBus, getMaxX, getMaxY, isActive, reset, setComBus,  
toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Field Detail

LINEAR_FORMATION_NS

```
public static final java.lang.String LINEAR_FORMATION_NS
```

Name space of the settings (append to group name space)

See Also:

[Constant Field Values](#)

START_LOCATION_S

```
public static final java.lang.String START_LOCATION_S
```

Per node group setting for defining the start coordinates of the line ("startLocation")

See Also:

[Constant Field Values](#)

END_LOCATION_S

```
public static final java.lang.String END_LOCATION_S
```

Per node group setting for defining the end coordinates of the line ("endLocation")

See Also:

[Constant Field Values](#)

Constructor Detail

LinearFormation

```
public LinearFormation(Settings s)
```

Creates a new movement model based on a Settings object's settings.

Parameters:

s - The Settings object where the settings are read from

LinearFormation

```
public LinearFormation(LinearFormation lf)
```

Copy constructor.

Parameters:

`lf` - The LinearFormation prototype

Method Detail

getInitialLocation

```
public Coord getInitialLocation()
```

Returns the the location of the node in the formation

Specified by:

[getInitialLocation](#) in class [MovementModel](#)

Returns:

the the location of the node in the formation

getPath

```
public Path getPath()
```

Returns a single coordinate path (using the only possible coordinate)

Specified by:

[getPath](#) in class [MovementModel](#)

Returns:

a single coordinate path

nextPathAvailable

```
public double nextPathAvailable()
```

Returns Double.MAX_VALUE (no paths available)

Overrides:

[nextPathAvailable](#) in class [MovementModel](#)

Returns:

The sim time when node should ask the next time for a path

replicate

```
public LinearFormation replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Specified by:

[replicate](#) in class [MovementModel](#)

Returns:

A new movement model with the same settings as this model

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

gui

Class MainWindow

```

java.lang.Object
  ↘ java.awt.Component
    ↘ java.awt.Container
      ↘ java.awt.Window
        ↘ java.awt.Frame
          ↘ javax.swing.JFrame
            ↘ gui.MainWindow

```

All Implemented Interfaces:

`java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants`

```

public class MainWindow
extends javax.swing.JFrame

```

Main window for the program. Takes care of layouting the main components in the window.

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JFrame

`javax.swing.JFrame.AccessibleJFrame`

Nested classes/interfaces inherited from class java.awt.Frame

`java.awt.Frame.AccessibleAWTFrame`

Nested classes/interfaces inherited from class java.awt.Window

`java.awt.Window.AccessibleAWTWindow`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

Fields inherited from class javax.swing.JFrame

`accessibleContext, EXIT_ON_CLOSE, rootPane, rootPaneCheckingEnabled`

Fields inherited from class java.awt.Frame

CROSSHAIR_CURSOR, DEFAULT_CURSOR, E_RESIZE_CURSOR, HAND_CURSOR, ICONIFIED, MAXIMIZED_BOTH, MAXIMIZED_HORIZ, MAXIMIZED_VERT, MOVE_CURSOR, N_RESIZE_CURSOR, NE_RESIZE_CURSOR, NORMAL, NW_RESIZE_CURSOR, S_RESIZE_CURSOR, SE_RESIZE_CURSOR, SW_RESIZE_CURSOR, TEXT_CURSOR, W_RESIZE_CURSOR, WAIT_CURSOR

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface javax.swing.WindowConstants

DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

[MainWindow](#)(java.lang.String scenName, [World](#) world, [PlayField](#) field, [GUIControls](#) guiControls, [InfoPanel](#) infoPanel, [EventLogPanel](#) elp, [DTNSimGUI](#) gui)

Method Summary

javax.swing.JScrollPane

[getPlayFieldScroll\(\)](#)

Returns a reference of the play field scroll panel

Methods inherited from class javax.swing.JFrame

addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getGraphics, getJMenuBar, getLayeredPane, getRootPane, getTransferHandler, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled, paramString, processWindowEvent, remove, repaint, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update

Methods inherited from class java.awt.Frame

addNotify, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setCursor, setExtendedState, setMaximizedBounds, setMenuBar, setResizable, setState, setTitle, setUndecorated

Methods inherited from class java.awt.Window

addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getIconImages, getInputContext, getListeners, getLocale, getModalExclusionType, getMostRecentFocusOwner, getOwnedWindows, getOwner, getOwnerlessWindows, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindows, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isShowing, pack, postEvent, processEvent, processWindowFocusEvent, processWindowStateEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, reshape, setAlwaysOnTop, setBounds, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setIconImages, setLocationByPlatform, setLocationRelativeTo, setMinimumSize, setModalExclusionType, setSize, setSize, setVisible, show, toBack, toFront

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt,

```
findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents, getComponentZOrder, getContainerListeners,
getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize,
getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paint, paintComponents, preferredSize, print, printComponents,
processContainerEvent, remove, removeAll, removeContainerListener, setComponentZOrder,
setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont,
transferFocusBackward, transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, contains, createImage, createImage, createVolatileImage, createVolatileImage,
disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBaseline, getBaselineResizeBehavior, getBounds,
getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor,
getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics,
getForeground, getHeight, getHierarchyBoundsListeners, getHierarchyListeners,
getIgnoreRepaint, getInputMethodListeners, getInputMethodRequests, getKeyListeners,
getLocation, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners,
getMousePosition, getMouseWheelListeners, getName, getParent, getPeer,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getTreeLock,
getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isBackgroundSet,
isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner,
isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet,
isMinimumSizeSet, isOpaque, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp, list,
list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove,
mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll,
processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent,
processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent,
processMouseWheelEvent, removeComponentListener, removeFocusListener,
removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener,
removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint,
requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resize, resize,
setBackground, setComponentOrientation, setDropTarget, setEnabled, setFocusable,
setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setLocation,
setLocation, setMaximumSize, setName, setPreferredSize, show, size, toString, transferFocus,
transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Methods inherited from interface java.awt.MenuContainer

```
getFont, postEvent
```

Constructor Detail

MainWindow

```
public MainWindow(java.lang.String scenName,
                 World world,
                 PlayField field,
                 GUIControls guiControls,
                 InfoPanel infoPanel,
                 EventLogPanel elp,
                 DTNSimGUI gui)
```

Method Detail

getPlayFieldScroll

```
public javax.swing.JScrollPane getPlayFieldScroll()
```

Returns a reference of the play field scroll panel

Returns:

a reference of the play field scroll panel

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement

Class MapBasedMovement

```
java.lang.Object
  └── movement.MovementModel
      └── movement.MapBasedMovement
```

All Implemented Interfaces:

[SwitchableMovement](#)

Direct Known Subclasses:

[BusTravellerMovement](#), [CarMovement](#), [EveningActivityMovement](#), [HomeActivityMovement](#),
[MapRouteMovement](#), [OfficeActivityMovement](#), [ShortestPathMapBasedMovement](#)

```
public class MapBasedMovement
extends MovementModel
implements SwitchableMovement
```

Map based movement model which gives out Paths that use the roads of a SimMap.

Field Summary

protected boolean	backAllowed May a node choose to move back the same way it came at a crossing
static java.lang.String	FILE_S map file -setting id ("mapFile")
protected MapNode	lastMapNode node where the last path ended or node next to initial placement
static java.lang.String	MAP_BASE_MOVEMENT_NS map based movement model's settings namespace ("MapBasedMovement")
static java.lang.String	MAP_SELECT_S Per node group setting for selecting map node types that are OK for this node group to traverse trough.
protected int	maxPathLength max nrof map nodes to travel/path
protected int	minPathLength min nrof map nodes to travel/path
static java.lang.String	NROF_FILES_S number of map files -setting id ("nrofMapFiles")

Fields inherited from class movement.MovementModel

[comBus](#), [DEF_SPEEDS](#), [DEF_WAIT_TIMES](#), [maxSpeed](#), [maxWaitTime](#), [minSpeed](#), [minWaitTime](#),
[MOVEMENT_MODEL_NS](#), [rng](#), [RNG_SEED](#), [SPEED](#), [WAIT_TIME](#), [WORLD_SIZE](#)

Constructor Summary

protected

	<code>MapBasedMovement(MapBasedMovement mbm)</code> Copyconstructor.
	<code>MapBasedMovement(Settings settings)</code> Creates a new MapBasedMovement based on a Settings object's settings.
	<code>MapBasedMovement(Settings settings, SimMap newMap, int nrofMaps)</code> Creates a new MapBasedMovement based on a Settings object's settings but with different SimMap

Method Summary

<code>Coord</code>	<code>getInitialLocation()</code> Returns a (random) coordinate that is between two adjacent MapNodes
<code>Coord</code>	<code>getLastLocation()</code> Get the last location the getPath() of this movement model has returned
<code>SimMap</code>	<code>getMap()</code> Returns the SimMap this movement model uses
<code>protected int[]</code>	<code>getOkMapNodeTypes()</code> Returns map node types that are OK for this movement model in an array or null if all values are considered ok
<code>Path</code>	<code>getPath()</code> Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
<code>boolean</code>	<code>isReady()</code> Checks if the movement model is finished doing its task and it's time to switch to the next movement model.
<code>MapBasedMovement</code>	<code>replicate()</code> Creates a replicate of the movement model.
<code>protected MapNode</code>	<code>selectRandomOkNode(java.util.List<MapNode> nodes)</code> Selects and returns a random node that is OK from a list of nodes.
<code>void</code>	<code>setLocation(Coord lastWaypoint)</code> Tell the movement model what its current location is

Methods inherited from class movement.MovementModel

`generateSpeed, generateWaitTime, getComBus, getMaxX, getMaxY, isActive, nextPathAvailable, reset, setComBus, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

lastMapNode

`protected MapNode lastMapNode`

node where the last path ended or node next to initial placement

maxPathLength

```
protected int maxPathLength  
    max nrof map nodes to travel/path
```

minPathLength

```
protected int minPathLength  
    min nrof map nodes to travel/path
```

backAllowed

```
protected boolean backAllowed  
    May a node choose to move back the same way it came at a crossing
```

MAP_BASE_MOVEMENT_NS

```
public static final java.lang.String MAP_BASE_MOVEMENT_NS  
    map based movement model's settings namespace ("MapBasedMovement")
```

See Also:

[Constant Field Values](#)

NROF_FILES_S

```
public static final java.lang.String NROF_FILES_S  
    number of map files -setting id ("nrofMapFiles")
```

See Also:

[Constant Field Values](#)

FILE_S

```
public static final java.lang.String FILE_S  
    map file -setting id ("mapFile")
```

See Also:

[Constant Field Values](#)

MAP_SELECT_S

```
public static final java.lang.String MAP_SELECT_S  
    Per node group setting for selecting map node types that are OK for this node group to traverse through.  
    Value must be a comma separated list of integers in range of [1,31]. Values reference to map file indexes  
    (see FILE\_S). If setting is not defined, all map nodes are considered OK.
```

See Also:

[Constant Field Values](#)

Constructor Detail

MapBasedMovement

```
public MapBasedMovement(Settings settings)
```

Creates a new MapBasedMovement based on a Settings object's settings.

Parameters:

settings - The Settings object where the settings are read from

MapBasedMovement

```
public MapBasedMovement(Settings settings,
                      SimMap newMap,
                      int nrofMaps)
```

Creates a new MapBasedMovement based on a Settings object's settings but with different SimMap

Parameters:

settings - The Settings object where the settings are read from

newMap - The SimMap to use

nrofMaps - How many map "files" are in the map

MapBasedMovement

```
protected MapBasedMovement(MapBasedMovement mbm)
```

Copyconstructor.

Parameters:

mbm - The MapBasedMovement object to base the new object to

Method Detail

getInitialLocation

```
public Coord getInitialLocation()
```

Returns a (random) coordinate that is between two adjacent MapNodes

Specified by:

[getInitialLocation](#) in class [MovementModel](#)

Returns:

The initial coordinates for a node

getOkMapNodeTypes

```
protected int[] getOkMapNodeTypes()
```

Returns map node types that are OK for this movement model in an array or null if all values are considered

ok

Returns:

map node types that are OK for this movement model in an array

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Specified by:

[getPath](#) in class [MovementModel](#)

Returns:

A new path or null

selectRandomOkNode

```
protected MapNode selectRandomOkNode(java.util.List<MapNode> nodes)
```

Selects and returns a random node that is OK from a list of nodes. Whether node is OK, is determined by the okMapNodeTypes list. If okMapNodeTypes are defined, the given list **must** contain at least one OK node to prevent infinite looping.

Parameters:

nodes - The list of nodes to choose from.

Returns:

A random node from the list (that is OK if ok list is defined)

getMap

```
public SimMap getMap()
```

Returns the SimMap this movement model uses

Returns:

The SimMap this movement model uses

replicate

```
public MapBasedMovement replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Specified by:

[replicate](#) in class [MovementModel](#)

Returns:

A new movement model with the same settings as this model

getLastLocation

```
public Coord getLastLocation()
```

Description copied from interface: [SwitchableMovement](#)

Get the last location the getPath() of this movement model has returned

Specified by:

[getLastLocation](#) in interface [SwitchableMovement](#)

Returns:

the last location

setLocation

```
public void setLocation(Coord lastWaypoint)
```

Description copied from interface: [SwitchableMovement](#)

Tell the movement model what its current location is

Specified by:

[setLocation](#) in interface [SwitchableMovement](#)

isReady

```
public boolean isReady()
```

Description copied from interface: [SwitchableMovement](#)

Checks if the movement model is finished doing its task and it's time to switch to the next movement model. The method should be called between getPath() calls.

Specified by:

[isReady](#) in interface [SwitchableMovement](#)

Returns:

true if ready

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

gui.playfield

Class MapGraphic

```
java.lang.Object
└ gui.playfield.PlayFieldGraphic
    └ gui.playfield.MapGraphic
```

```
public class MapGraphic
extends PlayFieldGraphic
```

PlayfieldGraphic for SimMap visualization

Field Summary

Fields inherited from class [gui.playfield.PlayFieldGraphic](#)

[scale](#)

Constructor Summary

[MapGraphic](#)([SimMap](#) simMap)

Method Summary

void [draw](#)([java.awt.Graphics2D](#) g2)

Draws the graphic component to the graphics context g2

Methods inherited from class [gui.playfield.PlayFieldGraphic](#)

[getScale](#), [invScale](#), [scale](#), [scale](#), [setScale](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

MapGraphic

```
public MapGraphic(SimMap simMap)
```

Method Detail

draw

```
public void draw(java.awt.Graphics2D g2)
```

Description copied from class: [PlayFieldGraphic](#)

Draws the graphic component to the graphics context g2

Specified by:

[draw](#) in class [PlayFieldGraphic](#)

Parameters:

g2 - The context to draw the graphics to

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement.map

Class MapNode

```
java.lang.Object
└ movement.map.MapNode
```

All Implemented Interfaces:

`java.lang.Comparable<MapNode>`

```
public class MapNode
extends java.lang.Object
implements java.lang.Comparable<MapNode>
```

A node in a SimMap. Node has a location, 0-n neighbors that it is connected to and possibly a type identifier.

Field Summary

static int	<u>MAX_TYPE</u> Biggest valid type of a node: 31
static int	<u>MIN_TYPE</u> Smallest valid type of a node: 1

Constructor Summary

[MapNode](#)(`Coord` location)

Constructor.

Method Summary

void	<u>addNeighbor</u> (<code>MapNode</code> node) Adds the node as this node's neighbour (unless the node is null)
void	<u>addType</u> (int type) Adds a type indicator to this node
int	<u>compareTo</u> (<code>MapNode</code> o) Compares two map nodes by their coordinates
<code>Coord</code>	<u>getLocation</u> () Returns the location of the node
<code>java.util.List<MapNode></code>	<u>getNeighbors</u> () Returns the neighbors of this node.
boolean	<u>isType</u> (int type) Returns true if this node is of given type, false if none of node's type(s) match to given type or node doesn't have type at all
boolean	<u>isType</u> (int[] types) Returns true if the node's types match any of the given types
<code>java.lang.String</code>	<u>toString</u> ()

|| Returns a String representation of the map node ||

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

MIN_TYPE

`public static final int MIN_TYPE`

Smallest valid type of a node: 1

See Also:

[Constant Field Values](#)

MAX_TYPE

`public static final int MAX_TYPE`

Biggest valid type of a node: 31

See Also:

[Constant Field Values](#)

Constructor Detail

MapNode

`public MapNode(Coord location)`

Constructor. Creates a map node to a location.

Parameters:

`location` - The location of the node.

Method Detail

addType

`public void addType(int type)`

Adds a type indicator to this node

Parameters:

`type` - An integer from range [1, 31]

isType

`public boolean isType(int type)`

Returns true if this node is of given type, false if none of node's type(s) match to given type or node doesn't have type at all

Parameters:

`type` - The type (integer from range [[1](#), [31](#)])

Returns:

True if this node is of given type

isType

```
public boolean isType(int[] types)
```

Returns true if the node's types match any of the given types

Parameters:

`types` - The types to check (array of values in range [[1](#), [31](#)])

Returns:

True if at least one of the types matched, false if none of the types matched

See Also:

[isType\(int\)](#)

addNeighbor

```
public void addNeighbor(MapNode node)
```

Adds the node as this node's neighbour (unless the node is null)

Parameters:

`node` - The node to add or null for no action

getLocation

```
public Coord getLocation()
```

Returns the location of the node

Returns:

the location of the node

getNeighbors

```
public java.util.List<MapNode> getNeighbors()
```

Returns the neighbors of this node.

Returns:

the neighbors in a list

toString

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

Returns a String representation of the map node

Overrides:

toString in class java.lang.Object

Returns:

a String representation of the map node

compareTo

```
public int compareTo(MapNode o)
```

Compares two map nodes by their coordinates

Specified by:

compareTo in interface java.lang.Comparable<MapNode>

Parameters:

- o - The other MapNode
-

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement.map

Class MapRoute

```
java.lang.Object
└ movement.map.MapRoute
```

```
public class MapRoute
extends java.lang.Object
```

A route that consists of map nodes. There can be different kind of routes and the type is determined by the type parameter ([1](#) or [2](#)).

Field Summary

static int	CIRCULAR Type of the route ID: circular (1).
static int	PINGPONG Type of the route ID: ping-pong (2).

Constructor Summary

[MapRoute](#)(int type, java.util.List<[MapNode](#)> stops)

Creates a new map route

Method Summary

int	getNrofStops() Returns the number of stops on this route
java.util.List< MapNode >	getStops()
MapNode	nextStop() Returns the next stop on the route (depending on the route mode)
static java.util.List< MapRoute >	readRoutes (java.lang.String fileName, int type, SimMap map) Reads routes from files defined in Settings
MapRoute	replicate() Returns a new route with the same settings
void	setNextIndex (int index) Sets the next index for this route
java.lang.String	toString()

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

CIRCULAR

```
public static final int CIRCULAR
```

Type of the route ID: circular (1). After reaching the last node on path, the next node is the first node

See Also:

[Constant Field Values](#)

PINGPONG

```
public static final int PINGPONG
```

Type of the route ID: ping-pong (2). After last node on path, the direction on path is reversed

See Also:

[Constant Field Values](#)

Constructor Detail

MapRoute

```
public MapRoute(int type,
                java.util.List<MapNode> stops)
```

Creates a new map route

Parameters:

stops - The stops of this route in a list
type - Type of the route (e.g. CIRCULAR or PINGPONG)

Method Detail

setNextIndex

```
public void setNextIndex(int index)
```

Sets the next index for this route

Parameters:

index - The index to set

getNrofStops

```
public int getNrofStops()
```

Returns the number of stops on this route

Returns:

the number of stops on this route

getStops

```
public java.util.List<MapNode> getStops()
```

nextStop

```
public MapNode nextStop()
```

Returns the next stop on the route (depending on the route mode)

Returns:

the next stop on the route

replicate

```
public MapRoute replicate()
```

Returns a new route with the same settings

Returns:

a replicate of this route

toString

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

Overrides:

`toString` in class `java.lang.Object`

readRoutes

```
public static java.util.List<MapRoute> readRoutes( java.lang.String fileName,
                           int type,
                           SimMap map)
```

Reads routes from files defined in Settings

Parameters:

`fileName` - name of the file where to read routes

`type` - Type of the route

`map` - SimMap where corresponding map nodes are found

Returns:

A list of MapRoutes that were read

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement

Class MapRouteMovement

```
java.lang.Object
  └── movement.MovementModel
      └── movement.MapBasedMovement
          └── movement.MapRouteMovement
```

All Implemented Interfaces:

[SwitchableMovement](#)

Direct Known Subclasses:

[BusMovement](#)

```
public class MapRouteMovement
extends MapBasedMovement
implements SwitchableMovement
```

Map based movement model that uses predetermined paths within the map area. Nodes using this model (can) stop on every route waypoint and find their way to next waypoint using [DijkstraPathFinder](#). There can be different type of routes; see [ROUTE_TYPE_S](#).

Field Summary

static java.lang.String	ROUTE_FILE_S Per node group setting used for selecting a route file ("routeFile")
static java.lang.String	ROUTE_FIRST_STOP_S Per node group setting for selecting which stop (counting from 0 from the start of the route) should be the first one.
static java.lang.String	ROUTE_TYPE_S Per node group setting used for selecting a route's type ("routeType").

Fields inherited from class movement.MapBasedMovement

```
backAllowed, FILE_S, lastMapNode, MAP_BASE_MOVEMENT_NS, MAP_SELECT_S, maxPathLength,
minPathLength, NROF_FILES_S
```

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

protected	MapRouteMovement(MapRouteMovement proto) Copyconstructor.
	MapRouteMovement(Settings settings) Creates a new movement model based on a Settings object's settings.

Method Summary

Coord	<code>getInitialLocation()</code> Returns the first stop on the route
Coord	<code>getLastLocation()</code> Get the last location the getPath() of this movement model has returned
Path	<code>getPath()</code> Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
<code>java.util.List<MapNode></code>	<code>getStops()</code> Returns the list of stops on the route
MapRouteMovement	<code>replicate()</code> Creates a replicate of the movement model.

Methods inherited from class movement.[MapBasedMovement](#)

[getMap](#), [getOkMapNodeType](#)s, [isReady](#), [selectRandomOkNode](#), [setLocation](#)

Methods inherited from class movement.[MovementModel](#)

[generateSpeed](#), [generateWaitTime](#), [getComBus](#), [getMaxX](#), [getMaxY](#), [isActive](#), [nextPathAvailable](#), [reset](#), [setComBus](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface movement.[SwitchableMovement](#)

[isReady](#), [setLocation](#)

Field Detail

ROUTE_FILE_S

public static final java.lang.String **ROUTE_FILE_S**

Per node group setting used for selecting a route file ("routeFile")

See Also:

[Constant Field Values](#)

ROUTE_TYPE_S

public static final java.lang.String **ROUTE_TYPE_S**

Per node group setting used for selecting a route's type ("routeType"). Integer value from [MapRoute](#) class.

See Also:

[Constant Field Values](#)

ROUTE_FIRST_STOP_S

```
public static final java.lang.String ROUTE_FIRST_STOP_S
```

Per node group setting for selecting which stop (counting from 0 from the start of the route) should be the first one. By default, or if a negative value is given, a random stop is selected.

See Also:

[Constant Field Values](#)

Constructor Detail

MapRouteMovement

```
public MapRouteMovement(Settings settings)
```

Creates a new movement model based on a Settings object's settings.

Parameters:

settings - The Settings object where the settings are read from

MapRouteMovement

```
protected MapRouteMovement(MapRouteMovement proto)
```

Copyconstructor. Gives a route to the new movement model from the list of routes and randomizes the starting position.

Parameters:

proto - The MapRouteMovement prototype

Method Detail

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Overrides:

[getPath](#) in class [MapBasedMovement](#)

Returns:

A new path or null

getInitialLocation

```
public Coord getInitialLocation()
```

Returns the first stop on the route

Overrides:

[getInitialLocation](#) in class [MapBasedMovement](#)

Returns:

The initial coordinates for a node

getLastLocation

```
public Coord getLastLocation()
```

Description copied from interface: [SwitchableMovement](#)

Get the last location the getPath() of this movement model has returned

Specified by:

[getLastLocation](#) in interface [SwitchableMovement](#)

Overrides:

[getLastLocation](#) in class [MapBasedMovement](#)

Returns:

the last location

replicate

```
public MapRouteMovement replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Overrides:

[replicate](#) in class [MapBasedMovement](#)

Returns:

A new movement model with the same settings as this model

getStops

```
public java.util.List<MapNode> getStops()
```

Returns the list of stops on the route

Returns:

The list of stops

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing.maxprop

Class MaxPropDijkstra

```
java.lang.Object
  ↴
  routing.maxprop.MaxPropDijkstra
```

```
public class MaxPropDijkstra
extends java.lang.Object
```

Dijkstra's shortest path implementation for MaxProp Router module.

Constructor Summary

[MaxPropDijkstra](#)(java.util.Map<java.lang.Integer, [MeetingProbabilitySet](#)> probs)

Constructor.

Method Summary

<pre>java.util.Map<java.lang.Integer, java.lang.Double></pre>	getCosts (java.lang.Integer from, java.util.Set<java.lang.Integer> to) Calculates total costs to the given set of target nodes.
---	---

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

MaxPropDijkstra

```
public MaxPropDijkstra(java.util.Map<java.lang.Integer, MeetingProbabilitySet> probs)
```

Constructor.

Parameters:

probs - A reference to the mapping of the known hosts meeting probability sets

Method Detail

getCosts

```
public java.util.Map<java.lang.Integer, java.lang.Double> getCosts(java.lang.Integer from,  

          java.util.Set<java.lang.Integer> to)
```

Calculates total costs to the given set of target nodes. The cost to a node is the sum of complements of probabilities that all the links come up as the next contact of the nodes.

Parameters:

from - The index (address) of the start node

to - The address set of destination nodes

Returns:

A map of (destination node, cost) tuples

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class MaxPropRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.MaxPropRouter
```

```
public class MaxPropRouter
extends ActiveRouter
```

Implementation of MaxProp router as described in *MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks* by John Burgess et al.

Field Summary

static java.lang.String	ALPHA_S The alpha parameter string
static int	BYTES_TRANSFERRED_AVG_SAMPLES Over how many samples the "average number of bytes transferred per transfer opportunity" is taken
static double	DEFAULT_ALPHA The default value for alpha
static int	DEFAULT_PROB_SET_MAX_SIZE Default value for the meeting probability set maximum size (50).
static java.lang.String	MAXPROP_NS Router's setting namespace ("MaxPropRouter")
static java.lang.String	PROB_SET_MAX_SIZE_S Meeting probability set maximum size -setting id ("probSetMaxSize").

Fields inherited from class routing.ActiveRouter

[DELETE_DELIVERED_S](#), [deleteDelivered](#), [RESPONSE_PREFIX](#), [sendingConnections](#), [TTL_CHECK_INTERVAL](#)

Fields inherited from class routing.MessageRouter

[B_SIZE_S](#), [DENIED_NO_SPACE](#), [DENIED_OLD](#), [DENIED_TTL](#), [DENIED_UNSPECIFIED](#), [MSG_TTL_S](#), [msgTtl](#), [O_MODE_FIFO](#), [O_MODE_RANDOM](#), [RCV_OK](#), [SEND_QUEUE_MODE_S](#), [TRY_LATER_BUSY](#)

Constructor Summary

protected	MaxPropRouter (MaxPropRouter r) Copy constructor.
	MaxPropRouter (Settings settings) Constructor.

Method Summary

int	<code>calcThreshold()</code>	Calculates and returns the current threshold value for the buffer's split based on the average number of bytes transferred per transfer opportunity and the hop counts of the messages in the buffer.
void	<code>changedConnection(Connection con)</code>	Called when a connection's state changes.
double	<code>getCost(DTNHost from, DTNHost to)</code>	Returns the message delivery cost between two hosts from this host's point of view.
protected <u>Message</u>	<code>getOldestMessage(boolean excludeMsgBeingSent)</code>	Returns the next message that should be dropped, according to MaxProp's message ordering scheme (see <code>MaxPropTupleComparator</code>).
<u>RoutingInfo</u>	<code>getRoutingInfo()</code>	Returns routing information about this router.
<u>Message</u>	<code>messageTransferred(java.lang.String id, DTNHost from)</code>	This method should be called (on the receiving host) after a message was successfully transferred.
<u>MessageRouter</u>	<code>replicate()</code>	Creates a replicate of this router.
protected void	<code>transferDone(Connection con)</code>	Method is called just before a transfer is finalized at <code>ActiveRouter.update()</code> .
void	<code>update()</code>	Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.[ActiveRouter](#)

[addToSendConnections](#), [canStartTransfer](#), [checkReceiving](#), [createNewMessage](#), [dropExpiredMessages](#), [exchangeDeliverableMessages](#), [getConnections](#), [getMessagesForConnected](#), [init](#), [isSending](#), [isTransferring](#), [makeRoomForMessage](#), [makeRoomForNewMessage](#), [receiveMessage](#), [requestDeliverableMessages](#), [shuffleMessages](#), [startTransfer](#), [transferAborted](#), [tryAllMessages](#), [tryAllMessagesToAllConnections](#), [tryMessagesForConnected](#), [tryMessagesToConnections](#)

Methods inherited from class routing.[MessageRouter](#)

[addApplication](#), [addToMessages](#), [compareByQueueMode](#), [deleteMessage](#), [getApplications](#), [getBufferSize](#), [getFreeBufferSize](#), [getHost](#), [getMessage](#), [getMessageCollection](#), [getNrofMessages](#), [hasMessage](#), [isDeliveredMessage](#), [isIncomingMessage](#), [messageAborted](#), [putToIncomingBuffer](#), [removeFromIncomingBuffer](#), [removeFromMessages](#), [sendMessage](#), [sortByQueueMode](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

MAXPROP_NS

public static final java.lang.String **MAXPROP_NS**

Router's setting namespace ("MaxPropRouter")

See Also:

[Constant Field Values](#)

PROB_SET_MAX_SIZE_S

```
public static final java.lang.String PROB_SET_MAX_SIZE_S
```

Meeting probability set maximum size -setting id ("probSetMaxSize"). The maximum amount of meeting probabilities to store.

See Also:

[Constant Field Values](#)

DEFAULT_PROB_SET_MAX_SIZE

```
public static final int DEFAULT_PROB_SET_MAX_SIZE
```

Default value for the meeting probability set maximum size (50).

See Also:

[Constant Field Values](#)

BYTES_TRANSFERRED_AVG_SAMPLES

```
public static int BYTES_TRANSFERRED_AVG_SAMPLES
```

Over how many samples the "average number of bytes transferred per transfer opportunity" is taken

ALPHA_S

```
public static final java.lang.String ALPHA_S
```

The alpha parameter string

See Also:

[Constant Field Values](#)

DEFAULT_ALPHA

```
public static final double DEFAULT_ALPHA
```

The default value for alpha

See Also:

[Constant Field Values](#)

Constructor Detail

MaxPropRouter

```
public MaxPropRouter(Settings settings)
```

Constructor. Creates a new prototype router based on the settings in the given Settings object.

Parameters:

settings - The settings object

MaxPropRouter

```
protected MaxPropRouter(MaxPropRouter r)
```

Copy constructor. Creates a new router based on the given prototype.

Parameters:

r - The router prototype where setting values are copied from

Method Detail

changedConnection

```
public void changedConnection(Connection con)
```

Description copied from class: [ActiveRouter](#)

Called when a connection's state changes. This version doesn't do anything but subclasses may want to override this.

Overrides:

[changedConnection](#) in class [ActiveRouter](#)

Parameters:

con - The connection that changed

messageTransferred

```
public Message messageTransferred(java.lang.String id,
                                    DTNHost from)
```

Description copied from class: [MessageRouter](#)

This method should be called (on the receiving host) after a message was successfully transferred. The transferred message is put to the message buffer unless this host is the final recipient of the message.

Overrides:

[messageTransferred](#) in class [ActiveRouter](#)

Parameters:

id - Id of the transferred message

from - Host the message was from (previous hop)

Returns:

The message that this host received

transferDone

```
protected void transferDone(Connection con)
```

Method is called just before a transfer is finalized at [ActiveRouter.update\(\)](#). MaxProp makes book keeping of the delivered messages so their IDs are stored.

Overrides:

[transferDone](#) in class [ActiveRouter](#)

Parameters:

con - The connection whose transfer was finalized

getOldestMessage

```
protected Message getOldestMessage(boolean excludeMsgBeingSent)
```

Returns the next message that should be dropped, according to MaxProp's message ordering scheme (see [MaxPropTupleComparator](#)).

Overrides:

[getOldestMessage](#) in class [ActiveRouter](#)

Parameters:

excludeMsgBeingSent - If true, excludes message(s) that are being sent from the next-to-be-dropped check (i.e., if next message to drop is being sent, the following message is returned)

Returns:

The oldest message or null if no message could be returned (no messages in buffer or all messages in buffer are being sent and excludeMsgBeingSent is true)

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

getCost

```
public double getCost(DTNHost from,  
                  DTNHost to)
```

Returns the message delivery cost between two hosts from this host's point of view. If there is no path between "from" and "to" host, Double.MAX_VALUE is returned. Paths are calculated only to hosts that this host has messages to.

Parameters:

from - The host where a message is coming from
to - The host where a message would be destined to

Returns:

The cost of the cheapest path to the destination or Double.MAX_VALUE if such a path doesn't exist

calcThreshold

```
public int calcThreshold()
```

Calculates and returns the current threshold value for the buffer's split based on the average number of bytes transferred per transfer opportunity and the hop counts of the messages in the buffer. Method is public only to make testing easier.

Returns:

current threshold value (hop count) for the buffer's split

getRoutingInfo

```
public RoutingInfo getRoutingInfo()
```

Description copied from class: [MessageRouter](#)

Returns routing information about this router.

Overrides:

[getRoutingInfo](#) in class [MessageRouter](#)

Returns:

The routing information.

replicate

```
public MessageRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class MaxPropRouterWithEstimation

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.MaxPropRouterWithEstimation
```

```
public class MaxPropRouterWithEstimation
extends ActiveRouter
```

Implementation of MaxProp router as described in *MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks* by John Burgess et al. but with parameter estimation for finding an alpha based on timescale definition: Extension of the protocol by adding a parameter alpha (default 1) By new connection, the delivery likelihood is increased by alpha and divided by 1+alpha. Using the default results in the original algorithm. Refer to Karvo and Ott, *Time Scales and Delay-Tolerant Routing Protocols* Chants, 2008 This version tries to estimate a good value of alpha from a timescale parameter given by the user, and from the encounters the node sees during simulation.

Field Summary

static int	BYTES TRANSFERRED AVG SAMPLES Over how many samples the "average number of bytes transferred per transfer opportunity" is taken
static double	DEFAULT ALPHA The default value for alpha
static java.lang.String	MAXPROP_NS MaxPROP router's setting namespace ("MaxPropRouterWithEstimation")
static java.lang.String	TIME SCALE S

Fields inherited from class routing.ActiveRouter

[DELETE DELIVERED S](#), [deleteDelivered](#), [RESPONSE PREFIX](#), [sendingConnections](#), [TTL CHECK INTERVAL](#)

Fields inherited from class routing.MessageRouter

[B SIZE S](#), [DENIED NO SPACE](#), [DENIED OLD](#), [DENIED TTL](#), [DENIED UNSPECIFIED](#), [MSG TTL S](#), [msgTtl](#), [O MODE FIFO](#), [O MODE RANDOM](#), [RCV OK](#), [SEND QUEUE MODE S](#), [TRY LATER BUSY](#)

Constructor Summary

protected	MaxPropRouterWithEstimation (MaxPropRouterWithEstimation r) Copy constructor.
	MaxPropRouterWithEstimation (Settings settings) Constructor.

Method Summary

int |

	<code>calcThreshold()</code> Calculates and returns the current threshold value for the buffer's split based on the average number of bytes transferred per transfer opportunity and the hop counts of the messages in the buffer.
<code>void</code>	<code>changedConnection(Connection con)</code> Called when a connection's state changes.
<code>double</code>	<code>getCost(DTNHost from, DTNHost to)</code> Returns the message delivery cost between two hosts from this host's point of view.
<code>protected Message</code>	<code>getOldestMessage(boolean excludeMsgBeingSent)</code> Returns the next message that should be dropped, according to MaxProp's message ordering scheme (see <code>MaxPropTupleComparator</code>).
<code>RoutingInfo</code>	<code>getRoutingInfo()</code> Returns routing information about this router.
<code>Message</code>	<code>messageTransferred(java.lang.String id, DTNHost from)</code> This method should be called (on the receiving host) after a message was successfully transferred.
<code>MessageRouter</code>	<code>replicate()</code> Creates a replicate of this router.
<code>protected void</code>	<code>transferDone(Connection con)</code> Method is called just before a transfer is finalized at <code>ActiveRouter.update()</code> .
<code>void</code>	<code>update()</code> Checks out all sending connections to finalize the ready ones and abort those whose connection went down.
<code>protected boolean</code>	<code>updateEstimators(DTNHost host)</code> Updates the MaxPROP estimators
<code>protected void</code>	<code>updateParam()</code> update the alpha parameter based on the estimators

Methods inherited from class routing.[ActiveRouter](#)

[addToSendingsConnections](#), [canStartTransfer](#), [checkReceiving](#), [createNewMessage](#), [dropExpiredMessages](#), [exchangeDeliverableMessages](#), [getConnections](#), [getMessagesForConnected](#), [init](#), [isSending](#), [isTransferring](#), [makeRoomForMessage](#), [makeRoomForNewMessage](#), [receiveMessage](#), [requestDeliverableMessages](#), [shuffleMessages](#), [startTransfer](#), [transferAborted](#), [tryAllMessages](#), [tryAllMessagesToAllConnections](#), [tryMessagesForConnected](#), [tryMessagesToConnections](#)

Methods inherited from class routing.[MessageRouter](#)

[addApplication](#), [addToMessages](#), [compareByQueueMode](#), [deleteMessage](#), [getApplications](#), [getBufferSize](#), [getFreeBufferSize](#), [getHost](#), [getMessage](#), [getMessageCollection](#), [getNrofMessages](#), [hasMessage](#), [isDeliveredMessage](#), [isIncomingMessage](#), [messageAborted](#), [putToIncomingBuffer](#), [removeFromIncomingBuffer](#), [removeFromMessages](#), [sendMessage](#), [sortByQueueMode](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#)

Field Detail

BYTES_TRANSFERRED_AVG_SAMPLES

```
public static int BYTES_TRANSFERRED_AVG_SAMPLES
```

Over how many samples the "average number of bytes transferred per transfer opportunity" is taken

MAXPROP_NS

```
public static final java.lang.String MAXPROP_NS
    MaxPROP router's setting namespace ("MaxPropRouterWithEstimation")
```

See Also:

[Constant Field Values](#)

TIME_SCALE_S

```
public static final java.lang.String TIME_SCALE_S
```

See Also:

[Constant Field Values](#)

DEFAULT_ALPHA

```
public static final double DEFAULT_ALPHA
```

The default value for alpha

See Also:

[Constant Field Values](#)

Constructor Detail

MaxPropRouterWithEstimation

```
public MaxPropRouterWithEstimation(Settings settings)
```

Constructor. Creates a new prototype router based on the settings in the given Settings object.

Parameters:

settings - The settings object

MaxPropRouterWithEstimation

```
protected MaxPropRouterWithEstimation(MaxPropRouterWithEstimation r)
```

Copy constructor. Creates a new router based on the given prototype.

Parameters:

r - The router prototype where setting values are copied from

Method Detail

changedConnection

```
public void changedConnection(Connection con)
```

Description copied from class: [ActiveRouter](#)

Called when a connection's state changes. This version doesn't do anything but subclasses may want to override this.

Overrides:

[changedConnection](#) in class [ActiveRouter](#)

Parameters:

con - The connection that changed

updateEstimators

```
protected boolean updateEstimators(DTNHost host)
```

Updates the MaxPROP estimators

Parameters:

host -

updateParam

```
protected void updateParam()
```

update the alpha parameter based on the estimators

messageTransferred

```
public Message messageTransferred(java.lang.String id,  
                                DTNHost from)
```

Description copied from class: [MessageRouter](#)

This method should be called (on the receiving host) after a message was successfully transferred. The transferred message is put to the message buffer unless this host is the final recipient of the message.

Overrides:

[messageTransferred](#) in class [ActiveRouter](#)

Parameters:

id - Id of the transferred message
from - Host the message was from (previous hop)

Returns:

The message that this host received

transferDone

```
protected void transferDone(Connection con)
```

Method is called just before a transfer is finalized at [ActiveRouter.update\(\)](#). MaxProp makes book keeping of the delivered messages so their IDs are stored.

Overrides:

[transferDone](#) in class [ActiveRouter](#)

Parameters:

con - The connection whose transfer was finalized

getOldestMessage

```
protected Message getOldestMessage(boolean excludeMsgBeingSent)
```

Returns the next message that should be dropped, according to MaxProp's message ordering scheme (see [MaxPropTupleComparator](#)).

Overrides:

[getOldestMessage](#) in class [ActiveRouter](#)

Parameters:

excludeMsgBeingSent - If true, excludes message(s) that are being sent from the next-to-be-dropped check (i.e., if next message to drop is being sent, the following message is returned)

Returns:

The oldest message or null if no message could be returned (no messages in buffer or all messages in buffer are being sent and excludeMsgBeingSent is true)

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

getCost

```
public double getCost(DTNHost from,  
                  DTNHost to)
```

Returns the message delivery cost between two hosts from this host's point of view. If there is no path between "from" and "to" host, Double.MAX_VALUE is returned. Paths are calculated only to hosts that this host has messages to.

Parameters:

from - The host where a message is coming from
to - The host where a message would be destined to

Returns:

The cost of the cheapest path to the destination or Double.MAX_VALUE if such a path doesn't exist

calcThreshold

```
public int calcThreshold()
```

Calculates and returns the current threshold value for the buffer's split based on the average number of bytes transferred per transfer opportunity and the hop counts of the messages in the buffer. Method is public only to make testing easier.

Returns:

current threshold value (hop count) for the buffer's split

getRoutingInfo

public [RoutingInfo](#) **getRoutingInfo()**

Description copied from class: [MessageRouter](#)

Returns routing information about this router.

Overrides:

[getRoutingInfo](#) in class [MessageRouter](#)

Returns:

The routing information.

replicate

public [MessageRouter](#) **replicate()**

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing.maxprop

Class MeetingProbabilitySet

```
java.lang.Object
└ routing.maxprop.MeetingProbabilitySet
```

```
public class MeetingProbabilitySet
extends java.lang.Object
```

Class for storing and manipulating the meeting probabilities for the MaxProp router module.

Field Summary

static int	INFINITE_SET_SIZE
------------	-----------------------------------

Constructor Summary

[MeetingProbabilitySet\(\)](#)

Constructor.

[MeetingProbabilitySet\(double alpha, java.util.List<java.lang.Integer> initiallyKnownNodes\)](#)

Constructor.

[MeetingProbabilitySet\(int maxSetSize, double alpha\)](#)

Constructor.

Method Summary

java.util.Map<java.lang.Integer, java.lang.Double>	getAllProbs() Returns a reference to the probability map of this probability set
double	getLastUpdateTime() Returns the time when this probability set was last updated
double	getProbFor(java.lang.Integer index) Returns the current delivery probability value for the given node index
MeetingProbabilitySet	replicate() Returns a deep copy of the probability set
void	setAlpha(double alpha) Enables changing the alpha parameter dynamically
java.lang.String	toString() Returns a String presentation of the probabilities
void	updateMeetingProbFor(java.lang.Integer index) Updates meeting probability for the given node index.

```
void updateMeetingProbFor(java.lang.Integer index,
                           double iet)
```

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail**INFINITE_SET_SIZE**

```
public static final int INFINITE_SET_SIZE
```

See Also:

[Constant Field Values](#)

Constructor Detail**MeetingProbabilitySet**

```
public MeetingProbabilitySet(int maxSetSize,
                           double alpha)
```

Constructor. Creates a probability set with empty node-probability mapping.

Parameters:

`maxSetSize` - Maximum size of the probability set; when the set is full, smallest values are dropped when new are added

MeetingProbabilitySet

```
public MeetingProbabilitySet()
```

Constructor. Creates a probability set with empty node-probability mapping and infinite set size

MeetingProbabilitySet

```
public MeetingProbabilitySet(double alpha,
                           java.util.List<java.lang.Integer> initiallyKnownNodes)
```

Constructor. Creates a probability set with equal probability for all the given node indexes.

Method Detail**updateMeetingProbFor**

```
public void updateMeetingProbFor(java.lang.Integer index)
```

Updates meeting probability for the given node index.

```
P(b) = P(b)_old + alpha
Normalize{P}
```

I.e., The probability of the given node index is increased by one and then all the probabilities are normalized so that their sum equals to 1.

Parameters:

index - The node index to update the probability for

updateMeetingProbFor

```
public void updateMeetingProbFor (java.lang.Integer index,
                                 double iet)
```

getProbFor

```
public double getProbFor (java.lang.Integer index)
```

Returns the current delivery probability value for the given node index

Parameters:

index - The index of the node to look the P for

Returns:

the current delivery probability value

getAllProbs

```
public java.util.Map<java.lang.Integer, java.lang.Double> getAllProbs()
```

Returns a reference to the probability map of this probability set

Returns:

a reference to the probability map of this probability set

getLastUpdateTime

```
public double getLastUpdateTime()
```

Returns the time when this probability set was last updated

Returns:

the time when this probability set was last updated

setAlpha

```
public void setAlpha (double alpha)
```

Enables changing the alpha parameter dynamically

replicate

```
public MeetingProbabilitySet replicate()
```

Returns a deep copy of the probability set

Returns:

a deep copy of the probability set

toString

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

Returns a String presentation of the probabilities

Overrides:

toString in class java.lang.Object

Returns:

a String presentation of the probabilities

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Class Message

```
java.lang.Object
└ core.Message
```

All Implemented Interfaces:

java.lang.Comparable<[Message](#)>

```
public class Message
extends java.lang.Object
implements java.lang.Comparable<Message>
```

A message that is created at a node or passed between nodes.

Field Summary

static int	INFINITE_TTL Value for infinite TTL of message
------------	---

Constructor Summary

[Message](#)([DTNHost](#) from, [DTNHost](#) to, java.lang.String id, int size)

Creates a new Message.

Method Summary

void	addNodeOnPath (DTNHost node) Adds a new node on the list of nodes this message has passed
void	addProperty (java.lang.String key, java.lang.Object value) Adds a generic property for this message.
int	compareTo (Message m) Compares two messages by their ID (alphabetically).
protected void	copyFrom (Message m) Deep copies message data from other message.
java.lang.String	getAppID ()
double	getCreationTime () Returns the time when this message was created
DTNHost	getFrom () Returns the node this message is originally from
int	getHopCount () Returns the amount of hops this message has passed
java.util.List< DTNHost >	getHops () Returns a list of nodes this message has passed so far

java.lang.String	getId()	Returns the ID of the message
java.lang.Object	getProperty(java.lang.String key)	Returns an object that was stored to this message using the given key.
double	getReceiveTime()	Returns the time when this message was received
Message	getRequest()	Returns the message this message is response to or null if this is not a response message
int	getResponseSize()	Returns the size of the requested response message or 0 if no response is requested.
int	getSize()	Returns the size of the message (in bytes)
DTNHost	getTo()	Returns the node this message is originally to
int	getTtl()	Returns the time to live (minutes) of the message or Integer.MAX_VALUE if the TTL is infinite.
int	getUniqueId()	Returns an ID that is unique per message instance (different for replicates too)
boolean	isResponse()	Returns true if this message is a response message
Message	replicate()	Returns a replicate of this message (identical except for the unique id)
static void	reset()	Resets all static fields to default values
void	setAppID(java.lang.String appID)	
void	setReceiveTime(double time)	Sets the time when this message was received.
void	setRequest(Message request)	If this message is a response to a request, sets the request message
void	setResponseSize(int size)	Sets the requested response message's size.
void	setTtl(int ttl)	Sets the initial TTL (time-to-live) for this message.
java.lang.String	toString()	Returns a string representation of the message
void	updateProperty(java.lang.String key, java.lang.Object value)	Updates a value for an existing property.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Field Detail

INFINITE_TTL

```
public static final int INFINITE_TTL
```

Value for infinite TTL of message

See Also:

[Constant Field Values](#)

Constructor Detail

Message

```
public Message(DTNHost from,
               DTNHost to,
               java.lang.String id,
               int size)
```

Creates a new Message.

Parameters:

- from - Who the message is (originally) from
- to - Who the message is (originally) to
- id - Message identifier (must be unique for message but will be the same for all replicates of the message)
- size - Size of the message (in bytes)

Method Detail

getFrom

```
public DTNHost getFrom()
```

Returns the node this message is originally from

Returns:

the node this message is originally from

getTo

```
public DTNHost getTo()
```

Returns the node this message is originally to

Returns:

the node this message is originally to

getId

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

Returns the ID of the message

Returns:

The message id

getUniqueId

```
public int getUniqueId()
```

Returns an ID that is unique per message instance (different for replicates too)

Returns:

The unique id

getSize

```
public int getSize()
```

Returns the size of the message (in bytes)

Returns:

the size of the message

addNodeOnPath

```
public void addNodeOnPath(DTNHost node)
```

Adds a new node on the list of nodes this message has passed

Parameters:

node - The node to add

getHops

```
public java.util.List<DTNHost> getHops()
```

Returns a list of nodes this message has passed so far

Returns:

The list as vector

getHopCount

```
public int getHopCount()
```

Returns the amount of hops this message has passed

Returns:

the amount of hops this message has passed

getTtl

```
public int getTtl()
```

Returns the time to live (minutes) of the message or Integer.MAX_VALUE if the TTL is infinite. Returned value can be negative if the TTL has passed already.

Returns:

The TTL (minutes)

setTtl

```
public void setTtl(int ttl)
```

Sets the initial TTL (time-to-live) for this message. The initial TTL is the TTL when the original message was created. The current TTL is calculated based on the time of

Parameters:

ttl - The time-to-live to set

setReceiveTime

```
public void setReceiveTime(double time)
```

Sets the time when this message was received.

Parameters:

time - The time to set

getReceiveTime

```
public double getReceiveTime()
```

Returns the time when this message was received

Returns:

The time

getCreationTime

```
public double getCreationTime()
```

Returns the time when this message was created

Returns:

the time when this message was created

setRequest

```
public void setRequest(Message request)
```

If this message is a response to a request, sets the request message

Parameters:

request - The request message

getRequest

```
public Message getRequest()
```

Returns the message this message is response to or null if this is not a response message

Returns:

the message this message is response to

isResponse

```
public boolean isResponse()
```

Returns true if this message is a response message

Returns:

true if this message is a response message

setResponseSize

```
public void setResponseSize(int size)
```

Sets the requested response message's size. If size == 0, no response is requested (default)

Parameters:

size - Size of the response message

getResponseSize

```
public int getResponseSize()
```

Returns the size of the requested response message or 0 if no response is requested.

Returns:

the size of the requested response message

toString

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

Returns a string representation of the message

Overrides:

toString in class java.lang.Object

Returns:

a string representation of the message

copyFrom

```
protected void copyFrom(Message m)
```

Deep copies message data from other message. If new fields are introduced to this class, most likely they should be copied here too (unless done in constructor).

Parameters:

m - The message where the data is copied

addProperty

```
public void addProperty(java.lang.String key,
                      java.lang.Object value)
                     throws SimError
```

Adds a generic property for this message. The key can be any string but it should be such that no other class accidentally uses the same value. The value can be any object but it's good idea to store only immutable objects because when message is replicated, only a shallow copy of the properties is made.

Parameters:

- key - The key which is used to lookup the value
- value - The value to store

Throws:

- [SimError](#) - if the message already has a value for the given key

getProperty

```
public java.lang.Object getProperty(java.lang.String key)
```

Returns an object that was stored to this message using the given key. If such object is not found, null is returned.

Parameters:

- key - The key used to lookup the object

Returns:

- The stored object or null if it isn't found

updateProperty

```
public void updateProperty(java.lang.String key,
                           java.lang.Object value)
                           throws SimError
```

Updates a value for an existing property. For storing the value first time, [addProperty\(String, Object\)](#) should be used which checks for name space clashes.

Parameters:

- key - The key which is used to lookup the value
- value - The new value to store

Throws:

- [SimError](#)

replicate

```
public Message replicate()
```

Returns a replicate of this message (identical except for the unique id)

Returns:

- A replicate of the message

compareTo

```
public int compareTo(Message m)
```

Compares two messages by their ID (alphabetically).

Specified by:

compareTo in interface java.lang.Comparable<[Message](#)>

See Also:

`String.compareTo(String)`

reset

`public static void reset()`

Resets all static fields to default values

getAppID

`public java.lang.String getAppID()`

Returns:

the appID

setAppID

`public void setAppID(java.lang.String appID)`

Parameters:

`appID` - the appID to set

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class MessageBurstGenerator

```
java.lang.Object
  └── input.MessageEventGenerator
      └── input.MessageBurstGenerator
```

All Implemented Interfaces:

[EventQueue](#)

```
public class MessageBurstGenerator
extends MessageEventGenerator
```

Message creation -external events generator. Creates bursts of messages where every source node (defined with [MessageEventGenerator.HOST_RANGE_S](#)) creates a new message to every destination node (defined with [MessageEventGenerator.TO_HOST_RANGE_S](#))on every interval. The message size, burst times, and inter-burst intervals can be configured like with [MessageEventGenerator](#).

See Also:

[MessageEventGenerator](#)

Field Summary

Fields inherited from class [input.MessageEventGenerator](#)

```
HOST RANGE S, hostRange, idPrefix, MESSAGE ID PREFIX S, MESSAGE INTERVAL S, MESSAGE SIZE S,
MESSAGE TIME S, msgTime, nextEventsTime, rng, TO HOST RANGE S, toHostRange
```

Constructor Summary

[MessageBurstGenerator\(Settings s\)](#)

Method Summary

ExternalEvent	nextEvent()
-------------------------------	-----------------------------

Returns the next message creation event

Methods inherited from class [input.MessageEventGenerator](#)

```
drawHostAddress, drawMessageSize, drawNextEventTimeDiff, drawToAddress, getID, nextEventsTime
```

Methods inherited from class [java.lang.Object](#)

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

MessageBurstGenerator

```
public MessageBurstGenerator(Settings s)
```

Method Detail

nextEvent

```
public ExternalEvent nextEvent()
```

Returns the next message creation event

Specified by:

[nextEvent](#) in interface [EventQueue](#)

Overrides:

[nextEvent](#) in class [MessageEventGenerator](#)

Returns:

The next event

See Also:

[EventQueue.nextEvent\(\)](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

input

Class MessageCreateEvent

```
java.lang.Object
  └── input.ExternalEvent
    └── input.MessageEvent
      └── input.MessageCreateEvent
```

All Implemented Interfaces:java.io.Serializable, java.lang.Comparable<[ExternalEvent](#)>

```
public class MessageCreateEvent
extends MessageEvent
```

External event for creating a message.

See Also:[Serialized Form](#)

Field Summary

Fields inherited from class input.MessageEvent

[fromAddr](#), [id](#), [toAddr](#)

Fields inherited from class input.ExternalEvent

[time](#)

Constructor Summary

```
MessageCreateEvent(int from, int to, java.lang.String id, int size, int responseSize,
double time)
```

Creates a message creation event with a optional response request

Method Summary

void	processEvent (World world)
------	---

Creates the message this event represents.

java.lang.String	toString ()
------------------	-----------------------------

Returns a String representation of the event

Methods inherited from class input.ExternalEvent

[compareTo](#), [getTime](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

MessageCreateEvent

```
public MessageCreateEvent(int from,
                          int to,
                          java.lang.String id,
                          int size,
                          int responseSize,
                          double time)
```

Creates a message creation event with a optional response request

Parameters:

- `from` - The creator of the message
- `to` - Where the message is destined to
- `id` - ID of the message
- `size` - Size of the message
- `responseSize` - Size of the requested response message or 0 if no response is requested
- `time` - Time, when the message is created

Method Detail

processEvent

```
public void processEvent(World world)
```

Creates the message this event represents.

Overrides:

[processEvent](#) in class [ExternalEvent](#)

Parameters:

- `world` - World where the actors of the event are

toString

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

Description copied from class: [ExternalEvent](#)

Returns a String representation of the event

Overrides:

[toString](#) in class [MessageEvent](#)

Returns:

- a String representation of the event

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class MessageDelayReport

```
java.lang.Object
└ report.Report
  └ report.MessageDelayReport
```

All Implemented Interfaces:

[MessageListener](#)

```
public class MessageDelayReport
extends Report
implements MessageListener
```

Reports delivered messages' delays (one line per delivered message) and cumulative delivery probability sorted by message delays. Ignores the messages that were created during the warm up period.

Field Summary

static java.lang.String	HEADER
-------------------------	------------------------

Fields inherited from class report.Report

DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING, PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime

Constructor Summary

[MessageDelayReport\(\)](#)

Constructor.

Method Summary

void	done() Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
void	init() Initializes the report output.
void	messageDeleted(Message m, DTNHost where, boolean dropped) Method is called when a message is deleted
void	messageTransferAborted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer was aborted before it finished
void	messageTransferred(Message m, DTNHost from, DTNHost to, boolean firstDelivery) Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer is started
void	

[newMessage \(Message m\)](#)

Method is called when a new message is created

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,
setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail**HEADER**

```
public static final java.lang.String HEADER
```

See Also:

[Constant Field Values](#)

Constructor Detail**MessageDelayReport**

```
public MessageDelayReport()
```

Constructor.

Method Detail**init**

```
public void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

newMessage

```
public void newMessage(Message m)
```

Description copied from interface: [MessageListener](#)

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

m - Message that was created

messageTransferred

```
public void messageTransferred(Message m,
                               DTNHost from,
                               DTNHost to,
                               boolean firstDelivery)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

m - The message that was transferred
 from - Node where the message was transferred from
 to - Node where the message was transferred to
 firstDelivery - Was the target node final destination of the message and received this message for the first time.

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

messageDeleted

```
public void messageDeleted(Message m,
                           DTNHost where,
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

m - The message that was deleted
 where - The host where the message was deleted
 dropped - True if the message was dropped, false if removed

messageTransferAborted

```
public void messageTransferAborted(Message m,
                                   DTNHost from,
                                   DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that was being transferred
`from` - Node where the message was being transferred from
`to` - Node where the message was being transferred to

messageTransferStarted

```
public void messageTransferStarted(Message m,  
                                 DTNHost from,  
                                 DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that is going to be transferred
`from` - Node where the message is transferred from
`to` - Node where the message is transferred to

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

input

Class MessageDeleteEvent

```
java.lang.Object
  └── input.ExternalEvent
    └── input.MessageEvent
      └── input.MessageDeleteEvent
```

All Implemented Interfaces:java.io.Serializable, java.lang.Comparable<[ExternalEvent](#)>

```
public class MessageDeleteEvent
extends MessageEvent
```

External event for deleting a message.

See Also:[Serialized Form](#)

Field Summary

Fields inherited from class input.MessageEvent[fromAddr](#), [id](#), [toAddr](#)**Fields inherited from class input.ExternalEvent**[time](#)

Constructor Summary

[MessageDeleteEvent](#)(int host, java.lang.String id, double time, boolean drop)

Creates a message delete event

Method Summary

void	processEvent (World world)
------	---

Deletes the message

java.lang.String	toString ()
------------------	-----------------------------

Returns a String representation of the event

Methods inherited from class input.ExternalEvent[compareTo](#), [getTime](#)**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

MessageDeleteEvent

```
public MessageDeleteEvent(int host,
                         java.lang.String id,
                         double time,
                         boolean drop)
```

Creates a message delete event

Parameters:

- host - Where to delete the message
- id - ID of the message
- time - Time when the message is deleted

Method Detail

processEvent

```
public void processEvent(World world)
```

Deletes the message

Overrides:

[processEvent](#) in class [ExternalEvent](#)

Parameters:

- world - World where the actors of the event are

toString

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

Description copied from class: [ExternalEvent](#)

Returns a String representation of the event

Overrides:

[toString](#) in class [MessageEvent](#)

Returns:

a String representation of the event

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class MessageDeliveryReport

```
java.lang.Object
└ report.Report
    └ report.MessageDeliveryReport
```

All Implemented Interfaces:

[MessageListener](#)

```
public class MessageDeliveryReport
extends Report
implements MessageListener
```

Report for of amount of messages delivered vs. time. A new report line is created every time when either a message is created or delivered. Messages created during the warm up period are ignored. For output syntax, see [HEADER](#).

Field Summary

static java.lang.String	HEADER
-------------------------	------------------------

Fields inherited from class report.Report

DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING, PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime

Constructor Summary

[MessageDeliveryReport\(\)](#)

Constructor.

Method Summary

void	done() Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
void	init() Initializes the report output.
void	messageDeleted(Message m, DTNHost where, boolean dropped) Method is called when a message is deleted
void	messageTransferAborted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer was aborted before it finished
void	messageTransferred(Message m, DTNHost from, DTNHost to, boolean firstDelivery) Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer is started

void	newMessage (Message m)
	Method is called when a new message is created

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,
setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

HEADER

```
public static java.lang.String HEADER
```

Constructor Detail

MessageDeliveryReport

```
public MessageDeliveryReport()
```

Constructor.

Method Detail

init

```
public void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

messageTransferred

```
public void messageTransferred(Message m,
                               DTNHost from,
                               DTNHost to,
                               boolean firstDelivery)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

m - The message that was transferred

`from` - Node where the message was transferred from
`to` - Node where the message was transferred to
`firstDelivery` - Was the target node final destination of the message and received this message for the first time.

newMessage

```
public void newMessage(Message m)
```

Description copied from interface: [MessageListener](#)

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

`m` - Message that was created

messageDeleted

```
public void messageDeleted(Message m,  
                           DTNHost where,  
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that was deleted
`where` - The host where the message was deleted
`dropped` - True if the message was dropped, false if removed

messageTransferAborted

```
public void messageTransferAborted(Message m,  
                                  DTNHost from,  
                                  DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that was being transferred
`from` - Node where the message was being transferred from
`to` - Node where the message was being transferred to

messageTransferStarted

```
public void messageTransferStarted(Message m,  
                                  DTNHost from,
```

[DTNHost](#) to)

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

- m - The message that is going to be transferred
- from - Node where the message is transferred from
- to - Node where the message is transferred to

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class MessageEvent

```
java.lang.Object
  └── input.ExternalEvent
      └── input.MessageEvent
```

All Implemented Interfaces:java.io.Serializable, java.lang.Comparable<[ExternalEvent](#)>**Direct Known Subclasses:**[MessageCreateEvent](#), [MessageDeleteEvent](#), [MessageRelayEvent](#)

```
public abstract class MessageEvent
extends ExternalEvent
```

A message related external event

See Also:[Serialized Form](#)

Field Summary

protected int	fromAddr address of the node the message is from
protected java.lang.String	id identifier of the message
protected int	toAddr address of the node the message is to

Fields inherited from class [input.ExternalEvent](#)

[time](#)

Constructor Summary

[MessageEvent](#)(int from, int to, java.lang.String id, double time)

Creates a message event

Method Summary

java.lang.String	toString() Returns a String representation of the event
------------------	--

Methods inherited from class [input.ExternalEvent](#)

[compareTo](#), [getTime](#), [processEvent](#)

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Field Detail**fromAddr**

```
protected int fromAddr
```

address of the node the message is from

toAddr

```
protected int toAddr
```

address of the node the message is to

id

```
protected java.lang.String id
```

identifier of the message

Constructor Detail**MessageEvent**

```
public MessageEvent(int from,
                    int to,
                    java.lang.String id,
                    double time)
```

Creates a message event

Parameters:

- from - Where the message comes from
- to - Who the message goes to
- id - ID of the message
- time - Time when the message event occurs

Method Detail**toString**

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

Description copied from class: [ExternalEvent](#)

Returns a String representation of the event

Overrides:

[toString](#) in class [ExternalEvent](#)

Returns:

a String representation of the event

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class MessageEventGenerator

```
java.lang.Object
└─input.MessageEventGenerator
```

All Implemented Interfaces:

[EventQueue](#)

Direct Known Subclasses:

[MessageBurstGenerator](#), [OneFromEachMessageGenerator](#), [OneToEachMessageGenerator](#)

```
public class MessageEventGenerator
extends java.lang.Object
implements EventQueue
```

Message creation -external events generator. Creates uniformly distributed message creation patterns whose message size and inter-message intervals can be configured.

Field Summary

static java.lang.String	HOST_RANGE_S Sender/receiver address range -setting id ("hosts").
protected int[]	hostRange Range of host addresses that can be senders or receivers
protected java.lang.String	idPrefix Prefix for the messages
static java.lang.String	MESSAGE_ID_PREFIX_S Message ID prefix -setting id ("prefix").
static java.lang.String	MESSAGE_INTERVAL_S Message creation interval range -setting id ("interval").
static java.lang.String	MESSAGE_SIZE_S Message size range -setting id ("size").
static java.lang.String	MESSAGE_TIME_S Message creation time range -setting id ("time").
protected double[]	msgTime Time range for message creation (min, max)
protected double	nextEventsTime Time of the next event (simulated seconds)
protected java.util.Random	rng Random number generator for this Class
static java.lang.String	TO_HOST_RANGE_S (Optional) receiver address range -setting id ("tohosts").
protected int[]	toHostRange Range of host addresses that can be receivers

Constructor Summary

[MessageEventGenerator \(Settings s\)](#)

Constructor, initializes the interval between events, and the size of messages generated, as well as number of hosts in the network.

Method Summary

protected int	drawHostAddress (int[] hostRange) Draws a random host address from the configured address range
protected int	drawMessageSize () Generates a (random) message size
protected int	drawNextEventTimeDiff () Generates a (random) time difference between two events
protected int	drawToAddress (int[] hostRange, int from) Draws a destination host address that is different from the "from" address
protected java.lang.String	getID () Returns a next free message ID
ExternalEvent	nextEvent () Returns the next message creation event
double	nextEventsTime () Returns next message creation event's time

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

MESSAGE_SIZE_S

public static final java.lang.String **MESSAGE_SIZE_S**

Message size range -setting id ("size"). Can be either a single value or a range (min, max) of uniformly distributed random values. Defines the message size (bytes).

See Also:

[Constant Field Values](#)

MESSAGE_INTERVAL_S

public static final java.lang.String **MESSAGE_INTERVAL_S**

Message creation interval range -setting id ("interval"). Can be either a single value or a range (min, max) of uniformly distributed random values. Defines the inter-message creation interval (seconds).

See Also:

[Constant Field Values](#)

HOST_RANGE_S

```
public static final java.lang.String HOST_RANGE_S
```

Sender/receiver address range -setting id ("hosts"). The lower bound is inclusive and upper bound exclusive.

See Also:

[Constant Field Values](#)

TO_HOST_RANGE_S

```
public static final java.lang.String TO_HOST_RANGE_S
```

(Optional) receiver address range -setting id ("tohosts"). If a value for this setting is defined, the destination hosts are selected from this range and the source hosts from the [HOST_RANGE_S](#) setting's range. The lower bound is inclusive and upper bound exclusive.

See Also:

[Constant Field Values](#)

MESSAGE_ID_PREFIX_S

```
public static final java.lang.String MESSAGE_ID_PREFIX_S
```

Message ID prefix -setting id ("prefix"). The value must be unique for all message sources, so if you have more than one message generator, use different prefix for all of them. The random number generator's seed is derived from the prefix, so by changing the prefix, you'll get also a new message sequence.

See Also:

[Constant Field Values](#)

MESSAGE_TIME_S

```
public static final java.lang.String MESSAGE_TIME_S
```

Message creation time range -setting id ("time"). Defines the time range when messages are created. No messages are created before the first and after the second value. By default, messages are created for the whole simulation time.

See Also:

[Constant Field Values](#)

nextEventsTime

```
protected double nextEventsTime
```

Time of the next event (simulated seconds)

hostRange

```
protected int[] hostRange
```

Range of host addresses that can be senders or receivers

toHostRange

```
protected int[] toHostRange
```

Range of host addresses that can be receivers

idPrefix

```
protected java.lang.String idPrefix
```

Prefix for the messages

msgTime

```
protected double[] msgTime
```

Time range for message creation (min, max)

rng

```
protected java.util.Random rng
```

Random number generator for this Class

Constructor Detail

MessageEventGenerator

```
public MessageEventGenerator(Settings s)
```

Constructor, initializes the interval between events, and the size of messages generated, as well as number of hosts in the network.

Parameters:

s - Settings for this generator.

Method Detail

drawHostAddress

```
protected int drawHostAddress(int[] hostRange)
```

Draws a random host address from the configured address range

Parameters:

hostRange - The range of hosts

Returns:

A random host address

drawMessageSize

```
protected int drawMessageSize()
```

Generates a (random) message size

Returns:

message size

drawNextEventTimeDiff

```
protected int drawNextEventTimeDiff()
```

Generates a (random) time difference between two events

Returns:

the time difference

drawToAddress

```
protected int drawToAddress(int[] hostRange,  
                           int from)
```

Draws a destination host address that is different from the "from" address

Parameters:

hostRange - The range of hosts

from - the "from" address

Returns:

a destination address from the range, but different from "from"

nextEvent

```
public ExternalEvent nextEvent()
```

Returns the next message creation event

Specified by:

[nextEvent](#) in interface [EventQueue](#)

Returns:

The next event

See Also:

[EventQueue.nextEvent\(\)](#)

nextEventsTime

```
public double nextEventsTime()
```

Returns next message creation event's time

Specified by:

[nextEventsTime](#) in interface [EventQueue](#)

Returns:

Next event's time

See Also:

[EventQueue.nextEventsTime\(\)](#)

getID

```
protected java.lang.String getID()
```

Returns a next free message ID

Returns:

next globally unique message ID

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

gui.playfield

Class MessageGraphic

```
java.lang.Object
└─gui.playfield.PlayFieldGraphic
    └─gui.playfield.MessageGraphic
```

```
public class MessageGraphic
extends PlayFieldGraphic
```

Visualization of a message

Field Summary

Fields inherited from class [gui.playfield.PlayFieldGraphic](#)

[scale](#)

Constructor Summary

[MessageGraphic](#)([DTNHost](#) from, [DTNHost](#) to)

Method Summary

void	draw (java.awt.Graphics2D g2)
------	---

Draws the graphic component to the graphics context g2

Methods inherited from class [gui.playfield.PlayFieldGraphic](#)

[getScale](#), [invScale](#), [scale](#), [scale](#), [setScale](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

MessageGraphic

```
public MessageGraphic(DTNHost from,
                     DTNHost to)
```

Method Detail

draw

```
public void draw(java.awt.Graphics2D g2)
```

Description copied from class: [PlayFieldGraphic](#)

Draws the graphic component to the graphics context g2

Specified by:

[draw](#) in class [PlayFieldGraphic](#)

Parameters:

g2 - The context to draw the graphics to

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class MessageGraphvizReport

```
java.lang.Object
└ report.Report
  └ report.MessageGraphvizReport
```

All Implemented Interfaces:

[MessageListener](#)

```
public class MessageGraphvizReport
extends Report
implements MessageListener
```

Creates a graphviz compatible graph of messages that were passed. Messages created during the warm up period are ignored.

Field Summary

static java.lang.String	GRAPH_NAME
Name of the graphviz report ("msggraph")	

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[MessageGraphvizReport\(\)](#)

Constructor.

Method Summary

void	done()	Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
protected void	init()	Initializes the report output.
void	messageDeleted(Message m, DTNHost where, boolean dropped)	Method is called when a message is deleted
void	messageTransferAborted(Message m, DTNHost from, DTNHost to)	Method is called when a message's transfer was aborted before it finished
void	messageTransferred(Message m, DTNHost from, DTNHost to, boolean firstDelivery)	Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted(Message m, DTNHost from, DTNHost to)	Method is called when a message's transfer is started
void		

[newMessage \(Message m\)](#)

Method is called when a new message is created

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,
setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail**GRAPH_NAME**

```
public static final java.lang.String GRAPH_NAME
```

Name of the graphviz report ("msggraph")

See Also:

[Constant Field Values](#)

Constructor Detail**MessageGraphvizReport**

```
public MessageGraphvizReport()
```

Constructor.

Method Detail**init**

```
protected void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

newMessage

```
public void newMessage(Message m)
```

Description copied from interface: [MessageListener](#)

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

`m` - Message that was created

messageTransferred

```
public void messageTransferred(Message m,
                             DTNHost from,
                             DTNHost to,
                             boolean firstDelivery)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

`m` - The message that was transferred
`from` - Node where the message was transferred from
`to` - Node where the message was transferred to
`firstDelivery` - Was the target node final destination of the message and received this message for the first time.

messageDeleted

```
public void messageDeleted(Message m,
                           DTNHost where,
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that was deleted
`where` - The host where the message was deleted
`dropped` - True if the message was dropped, false if removed

messageTransferAborted

```
public void messageTransferAborted(Message m,
                                   DTNHost from,
                                   DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that was being transferred
`from` - Node where the message was being transferred from
`to` - Node where the message was being transferred to

messageTransferStarted

```
public void messageTransferStarted(Message m,
                                 DTNHost from,
                                 DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

m - The message that is going to be transferred
 from - Node where the message is transferred from
 to - Node where the message is transferred to

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Interface MessageListener

All Known Implementing Classes:

[CreatedMessagesReport](#), [DeliveredMessagesReport](#), [DistanceDelayReport](#), [DTN2Reporter](#), [EventLogPanel](#), [EventLogReport](#), [MessageDelayReport](#), [MessageDeliveryReport](#), [MessageGraphvizReport](#), [MessageReport](#), [MessageStatsReport](#)

```
public interface MessageListener
```

Interface for classes that want to be informed about messages between hosts

Method Summary

void	messageDeleted (Message m, DTNHost where, boolean dropped)
	Method is called when a message is deleted
void	messageTransferAborted (Message m, DTNHost from, DTNHost to)
	Method is called when a message's transfer was aborted before it finished
void	messageTransferred (Message m, DTNHost from, DTNHost to, boolean firstDelivery)
	Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted (Message m, DTNHost from, DTNHost to)
	Method is called when a message's transfer is started
void	newMessage (Message m)
	Method is called when a new message is created

Method Detail

[newMessage](#)

```
void newMessage(Message m)
```

Method is called when a new message is created

Parameters:

m - Message that was created

[messageTransferStarted](#)

```
void messageTransferStarted(Message m,
                           DTNHost from,
                           DTNHost to)
```

Method is called when a message's transfer is started

Parameters:

m - The message that is going to be transferred
 from - Node where the message is transferred from
 to - Node where the message is transferred to

messageDeleted

```
void messageDeleted(Message m,  

                    DTNHost where,  

                    boolean dropped)
```

Method is called when a message is deleted

Parameters:

m - The message that was deleted
 where - The host where the message was deleted
 dropped - True if the message was dropped, false if removed

messageTransferAborted

```
void messageTransferAborted(Message m,  

                            DTNHost from,  

                            DTNHost to)
```

Method is called when a message's transfer was aborted before it finished

Parameters:

m - The message that was being transferred
 from - Node where the message was being transferred from
 to - Node where the message was being transferred to

messageTransferred

```
void messageTransferred(Message m,  

                        DTNHost from,  

                        DTNHost to,  

                        boolean firstDelivery)
```

Method is called when a message is successfully transferred from a node to another.

Parameters:

m - The message that was transferred
 from - Node where the message was transferred from
 to - Node where the message was transferred to
 firstDelivery - Was the target node final destination of the message and received this message for the first time.

report

Class MessageLocationReport

```
java.lang.Object
└ report.Report
  └ report.MessageLocationReport
```

All Implemented Interfaces:

[UpdateListener](#)

```
public class MessageLocationReport
extends Report
implements UpdateListener
```

Message location report. Reports the location (coordinates) of messages. The messages that are reported and the reporting interval can be configured.

Field Summary

protected int	granularity value of the granularity setting
static java.lang.String	GRANULARITY Reporting granularity -setting id ("granularity").
protected double	lastUpdate time of last update
static java.lang.String	REPORTED MESSAGES Reported messages -setting id ("messages").
protected java.util.HashSet<java.lang.String>	reportedMessages Identifiers of the message which are reported

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,  
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[MessageLocationReport\(\)](#)

Constructor.

Method Summary

void	updated (java.util.List< DTNHost > hosts)
------	---

Creates a new snapshot of the message locations if "granularity" seconds have passed since the last snapshot.

Methods inherited from class report.Report

```
addWarmupID, done, format, getAverage, getIntAverage, getIntMedian, getMedian,  

getScenarioName, getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent,  

removeWarmupID, setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

Field Detail

GRANULARITY

```
public static final java.lang.String GRANULARITY
```

Reporting granularity -setting id ("granularity"). Defines the interval how often (seconds) a new snapshot of message locations is created

See Also:

[Constant Field Values](#)

REPORTED_MESSAGES

```
public static final java.lang.String REPORTED_MESSAGES
```

Reported messages -setting id ("messages"). Defines the IDs of the messages that are reported (comma separated list)

See Also:

[Constant Field Values](#)

granularity

```
protected final int granularity
```

value of the granularity setting

lastUpdate

```
protected double lastUpdate
```

time of last update

reportedMessages

```
protected java.util.HashSet<java.lang.String> reportedMessages
```

Identifiers of the message which are reported

Constructor Detail

MessageLocationReport

```
public MessageLocationReport()
```

Constructor. Reads the settings and initializes the report module.

Method Detail

updated

```
public void updated(java.util.List<DTNHost> hosts)
```

Creates a new snapshot of the message locations if "granularity" seconds have passed since the last snapshot.

Specified by:

[updated](#) in interface [UpdateListener](#)

Parameters:

hosts - All the hosts in the world

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class MessageRelayEvent

```
java.lang.Object
  └── input.ExternalEvent
    └── input.MessageEvent
      └── input.MessageRelayEvent
```

All Implemented Interfaces:java.io.Serializable, java.lang.Comparable<[ExternalEvent](#)>

```
public class MessageRelayEvent
extends MessageEvent
```

External event for all the stages of relaying a message between two hosts (start and possible abort or delivery).

See Also:[Serialized Form](#)

Field Summary

static int	ABORTED Message relay stage constant for aborted delivery
static int	SENDING Message relay stage constant for start of sending
static java.lang.String[]	STAGE_STRINGS Stage constant -> String representation mapping
static int	TRANSFERRED Message relay stage constant for ready delivery

Fields inherited from class input.MessageEvent

[fromAddr](#), [id](#), [toAddr](#)

Fields inherited from class input.ExternalEvent

[time](#)

Constructor Summary

[MessageRelayEvent](#)(int from, int to, java.lang.String id, double time, int stage)

Creates a message relaying event

Method Summary

void	processEvent (World world) Relays the message
java.lang.String	toString ()

	Returns a String representation of the event
--	--

Methods inherited from class input.ExternalEvent

[compareTo](#), [getTime](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

SENDING

public static final int **SENDING**

Message relay stage constant for start of sending

See Also:

[Constant Field Values](#)

TRANSFERRED

public static final int **TRANSFERRED**

Message relay stage constant for ready delivery

See Also:

[Constant Field Values](#)

ABORTED

public static final int **ABORTED**

Message relay stage constant for aborted delivery

See Also:

[Constant Field Values](#)

STAGE_STRINGS

public static final java.lang.String[] **STAGE_STRINGS**

Stage constant -> String representation mapping

Constructor Detail

MessageRelayEvent

```
public MessageRelayEvent(int from,
                      int to,
                      java.lang.String id,
                      double time,
                      int stage)
```

Creates a message relaying event

Parameters:

- `from` - Where the message comes from (at this hop)
- `to` - Who the message goes to (at this hop)
- `id` - ID of the message
- `time` - Time when this event happens
- `stage` - The stage of the event (SENDING, TRANSFERRED, or ABORTED)

Method Detail

processEvent

```
public void processEvent(World world)
```

Relays the message

Overrides:

[processEvent](#) in class [ExternalEvent](#)

Parameters:

- `world` - World where the actors of the event are

toString

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

Description copied from class: [ExternalEvent](#)

Returns a String representation of the event

Overrides:

[toString](#) in class [MessageEvent](#)

Returns:

a String representation of the event

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class MessageReport

```
java.lang.Object
  └ report.Report
    └ report.MessageReport
```

All Implemented Interfaces:

[MessageListener](#)

```
public class MessageReport
extends Report
implements MessageListener
```

Reports delivered messages report: message_id creation_time deliver_time (duplicate)

Field Summary

static java.lang.String	HEADER
-------------------------	------------------------

Fields inherited from class report.Report

DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING, PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime

Constructor Summary

[MessageReport\(\)](#)

Constructor.

Method Summary

void	done() Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
void	init() Initializes the report output.
void	messageDeleted(Message m, DTNHost where, boolean dropped) Method is called when a message is deleted
void	messageTransferAborted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer was aborted before it finished
void	messageTransferred(Message m, DTNHost from, DTNHost to, boolean firstDelivery) Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer is started
void	newMessage(Message m)

Method is called when a new message is created

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,  

getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,  

setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

HEADER

```
public static final java.lang.String HEADER
```

See Also:

[Constant Field Values](#)

Constructor Detail

MessageReport

```
public MessageReport()
```

Constructor.

Method Detail

init

```
public void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

newMessage

```
public void newMessage(Message m)
```

Description copied from interface: [MessageListener](#)

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

m - Message that was created

messageTransferred

```
public void messageTransferred(Message m,
                               DTNHost from,
                               DTNHost to,
                               boolean firstDelivery)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

- m - The message that was transferred
- from - Node where the message was transferred from
- to - Node where the message was transferred to
- firstDelivery - Was the target node final destination of the message and received this message for the first time.

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

messageDeleted

```
public void messageDeleted(Message m,
                           DTNHost where,
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

- m - The message that was deleted
- where - The host where the message was deleted
- dropped - True if the message was dropped, false if removed

messageTransferAborted

```
public void messageTransferAborted(Message m,
                                   DTNHost from,
                                   DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that was being transferred
`from` - Node where the message was being transferred from
`to` - Node where the message was being transferred to

messageTransferStarted

```
public void messageTransferStarted(Message m,  
                                 DTNHost from,  
                                 DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that is going to be transferred
`from` - Node where the message is transferred from
`to` - Node where the message is transferred to

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class MessageRouter

```
java.lang.Object
└ routing.MessageRouter
```

Direct Known Subclasses:[ActiveRouter](#), [PassiveRouter](#)

```
public abstract class MessageRouter
extends java.lang.Object
```

Superclass for message routers.

Field Summary

static java.lang.String	B_SIZE_S Message buffer size -setting id ("bufferSize").
static int	DENIED_NO_SPACE Receive return value for not enough space in the buffer for the msg
static int	DENIED_OLD Receive return value for an old (already received) message
static int	DENIED_TTL Receive return value for messages whose TTL has expired
static int	DENIED_UNSPECIFIED Receive return value for unspecified reason
static java.lang.String	MSG_TTL_S Message TTL -setting id ("msgTtl").
protected int	msgTtl TTL for all messages
static int	Q_MODE_FIFO Setting value for FIFO queue mode
static int	Q_MODE_RANDOM Setting value for random queue mode
static int	RCV_OK Receive return value for OK
static java.lang.String	SEND_QUEUE_MODE_S Message/fragment sending queue type -setting id ("sendQueue").
static int	TRY_LATER_BUSY Receive return value for busy receiver

Constructor Summary

protected	MessageRouter (MessageRouter r)
-----------	--

Copy-constructor.

	<code>MessageRouter(Settings s)</code> Constructor.
--	--

Method Summary

	<code>void addApplication(Application app)</code> Adds an application to the attached applications list.
	<code>protected void addToMessages(Message m, boolean newMessage)</code> Adds a message to the message buffer and informs message listeners about new message (if requested).
	<code>abstract void changedConnection(Connection con)</code> Informs the router about change in connections state.
	<code>protected int compareByQueueMode(Message m1, Message m2)</code> Gives the order of the two given messages as defined by the current queue mode
	<code>boolean createNewMessage(Message m)</code> Creates a new message to the router.
	<code>void deleteMessage(java.lang.String id, boolean drop)</code> Deletes a message from the buffer and informs message listeners about the event
<code>java.util.Collection<Application></code>	<code>getApplications(java.lang.String ID)</code> Returns all the applications that want to receive messages for the given application ID.
	<code>int getBufferSize()</code> Returns the size of the message buffer.
	<code>int getFreeBufferSize()</code> Returns the amount of free space in the buffer.
<code>protected DTNHost</code>	<code>getHost()</code> Returns the host this router is in
<code>protected Message</code>	<code>getMessage(java.lang.String id)</code> Returns a message by ID.
<code>java.util.Collection<Message></code>	<code>getMessageCollection()</code> Returns a reference to the messages of this router in collection.
	<code>int getNrofMessages()</code> Returns the number of messages this router has
<code>RoutingInfo</code>	<code>getRoutingInfo()</code> Returns routing information about this router.
<code>protected boolean</code>	<code>hasMessage(java.lang.String id)</code> Checks if this router has a message with certain id buffered.
	<code>void init(DTNHost host, java.util.List<MessageListener> mListeners)</code> Initializes the router; i.e.
<code>protected boolean</code>	<code>isDeliveredMessage(Message m)</code> Returns true if a full message with same ID as the given message has been received by this host as the final recipient (at least once).
<code>protected boolean</code>	<code>isIncomingMessage(java.lang.String id)</code> Returns true if a message with the given ID is one of the currently incoming messages, false if not

	void messageAborted (java.lang.String id, DTNHost from, int bytesRemaining)	This method should be called (on the receiving host) when a message transfer was aborted.
Message	messageTransferred (java.lang.String id, DTNHost from)	This method should be called (on the receiving host) after a message was successfully transferred.
protected void	putToIncomingBuffer (Message m, DTNHost from)	Puts a message to incoming messages buffer.
int	receiveMessage (Message m, DTNHost from)	Try to start receiving a message from another host.
protected Message	removeFromIncomingBuffer (java.lang.String id, DTNHost from)	Removes and returns a message with a certain ID from the incoming messages buffer or null if such message wasn't found.
protected Message	removeFromMessages (java.lang.String id)	Removes and returns a message from the message buffer.
abstract MessageRouter	replicate ()	Creates a replicate of this router.
boolean	requestDeliverableMessages (Connection con)	Requests for deliverable message from this router to be sent through a connection.
void	sendMessage (java.lang.String id, DTNHost to)	Start sending a message to another host.
protected java.util.List	sortByQueueMode (java.util.List list)	Sorts/shuffles the given list according to the current sending queue mode.
java.lang.String	toString ()	Returns a String presentation of this router
void	update ()	Updates router.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

B_SIZE_S

public static final java.lang.String **B_SIZE_S**

Message buffer size -setting id ("bufferSize"). Integer value in bytes.

See Also:

[Constant Field Values](#)

MSG_TTL_S

public static final java.lang.String **MSG_TTL_S**

Message TTL -setting id ("msgTtl"). Value is in minutes and must be an integer.

See Also:

[Constant Field Values](#)

SEND_QUEUE_MODE_S

```
public static final java.lang.String SEND_QUEUE_MODE_S
```

Message/fragment sending queue type -setting id ("sendQueue"). This setting affects the order the messages and fragments are sent if the routing protocol doesn't define any particular order (e.g, if more than one message can be sent directly to the final recipient). Valid values are

- 1 : random (message order is randomized every time; default option)
- 2 : FIFO (most recently received messages are sent last)

See Also:

[Constant Field Values](#)

Q_MODE_RANDOM

```
public static final int Q_MODE_RANDOM
```

Setting value for random queue mode

See Also:

[Constant Field Values](#)

Q_MODE_FIFO

```
public static final int Q_MODE_FIFO
```

Setting value for FIFO queue mode

See Also:

[Constant Field Values](#)

RCV_OK

```
public static final int RCV_OK
```

Receive return value for OK

See Also:

[Constant Field Values](#)

TRY_LATER_BUSY

```
public static final int TRY_LATER_BUSY
```

Receive return value for busy receiver

See Also:

[Constant Field Values](#)

DENIED_OLD

```
public static final int DENIED_OLD
```

Receive return value for an old (already received) message

See Also:

[Constant Field Values](#)

DENIED_NO_SPACE

```
public static final int DENIED_NO_SPACE
```

Receive return value for not enough space in the buffer for the msg

See Also:

[Constant Field Values](#)

DENIED_TTL

```
public static final int DENIED_TTL
```

Receive return value for messages whose TTL has expired

See Also:

[Constant Field Values](#)

DENIED_UNSPECIFIED

```
public static final int DENIED_UNSPECIFIED
```

Receive return value for unspecified reason

See Also:

[Constant Field Values](#)

msgTtl

```
protected int msgTtl
```

TTL for all messages

Constructor Detail

MessageRouter

```
public MessageRouter(Settings s)
```

Constructor. Creates a new message router based on the settings in the given Settings object. Size of the message buffer is read from [B_SIZE_S](#) setting. Default value is Integer.MAX_VALUE.

Parameters:

s - The settings object

MessageRouter

```
protected MessageRouter(MessageRouter r)
```

Copy-constructor.

Parameters:

r - Router to copy the settings from.

Method Detail

init

```
public void init(DTNHost host,
                 java.util.List<MessageListener> mListeners)
```

Initializes the router; i.e. sets the host this router is in and message listeners that need to be informed about message related events etc.

Parameters:

host - The host this router is in

mListeners - The message listeners

update

```
public void update()
```

Updates router. This method should be called (at least once) on every simulation interval to update the status of transfer(s).

changedConnection

```
public abstract void changedConnection(Connection con)
```

Informs the router about change in connections state.

Parameters:

con - The connection that changed

getMessage

```
protected Message getMessage(java.lang.String id)
```

Returns a message by ID.

Parameters:

id - ID of the message

Returns:

The message

hasMessage

```
protected boolean hasMessage(java.lang.String id)
```

Checks if this router has a message with certain id buffered.

Parameters:

`id` - Identifier of the message

Returns:

True if the router has message with this id, false if not

isDeliveredMessage

```
protected boolean isDeliveredMessage(Message m)
```

Returns true if a full message with same ID as the given message has been received by this host as the **final** recipient (at least once).

Parameters:

`m` - message we're interested of

Returns:

true if a message with the same ID has been received by this host as the final recipient.

getMessageCollection

```
public java.util.Collection<Message> getMessageCollection()
```

Returns a reference to the messages of this router in collection. **Note:** If there's a chance that some message(s) from the collection could be deleted (or added) while iterating through the collection, a copy of the collection should be made to avoid concurrent modification exceptions.

Returns:

a reference to the messages of this router in collection

getNrofMessages

```
public int getNrofMessages()
```

Returns the number of messages this router has

Returns:

How many messages this router has

getBufferSize

```
public int getBufferSize()
```

Returns the size of the message buffer.

Returns:

The size or Integer.MAX_VALUE if the size isn't defined.

getFreeBufferSize

```
public int getFreeBufferSize()
```

Returns the amount of free space in the buffer. May return a negative value if there are more messages in the buffer than should fit there (because of creating new messages).

Returns:

The amount of free space (Integer.MAX_VALUE if the buffer size isn't defined)

getHost

```
protected DTNHost getHost()
```

Returns the host this router is in

Returns:

The host object

sendMessage

```
public void sendMessage(java.lang.String id,  
                      DTNHost to)
```

Start sending a message to another host.

Parameters:

`id` - Id of the message to send
`to` - The host to send the message to

requestDeliverableMessages

```
public boolean requestDeliverableMessages(Connection con)
```

Requests for deliverable message from this router to be sent trough a connection.

Parameters:

`con` - The connection to send the messages trough

Returns:

True if this router started a transfer, false if not

receiveMessage

```
public int receiveMessage(Message m,  
                         DTNHost from)
```

Try to start receiving a message from another host.

Parameters:

`m` - Message to put in the receiving buffer
`from` - Who the message is from

Returns:

Value zero if the node accepted the message (RCV_OK), value less than zero if node rejected the message (e.g. DENIED_OLD), value bigger than zero if the other node should try later (e.g. TRY_LATER_BUSY).

messageTransferred

```
public Message messageTransferred(java.lang.String id,
                                     DTNHost from)
```

This method should be called (on the receiving host) after a message was successfully transferred. The transferred message is put to the message buffer unless this host is the final recipient of the message.

Parameters:

`id` - Id of the transferred message
`from` - Host the message was from (previous hop)

Returns:

The message that this host received

putToIncomingBuffer

```
protected void putToIncomingBuffer(Message m,
                                 DTNHost from)
```

Puts a message to incoming messages buffer. Two messages with the same ID are distinguished by the from host.

Parameters:

`m` - The message to put
`from` - Who the message was from (previous hop).

removeFromIncomingBuffer

```
protected Message removeFromIncomingBuffer(java.lang.String id,
                                         DTNHost from)
```

Removes and returns a message with a certain ID from the incoming messages buffer or null if such message wasn't found.

Parameters:

`id` - ID of the message
`from` - The host that sent this message (previous hop)

Returns:

The found message or null if such message wasn't found

isIncomingMessage

```
protected boolean isIncomingMessage(java.lang.String id)
```

Returns true if a message with the given ID is one of the currently incoming messages, false if not

Parameters:

`id` - ID of the message

Returns:

True if such message is incoming right now

addToMessages

```
protected void addToMessages(Message m,
                           boolean newMessage)
```

Adds a message to the message buffer and informs message listeners about new message (if requested).

Parameters:

`m` - The message to add

`newMessage` - If true, message listeners are informed about a new message, if false, nothing is informed.

removeFromMessages

```
protected Message removeFromMessages(java.lang.String id)
```

Removes and returns a message from the message buffer.

Parameters:

`id` - Identifier of the message to remove

Returns:

The removed message or null if message for the ID wasn't found

messageAborted

```
public void messageAborted(java.lang.String id,
                           DTNHost from,
                           int bytesRemaining)
```

This method should be called (on the receiving host) when a message transfer was aborted.

Parameters:

`id` - Id of the message that was being transferred

`from` - Host the message was from (previous hop)

`bytesRemaining` - Nrof bytes that were left before the transfer would have been ready; or -1 if the number of bytes is not known

createNewMessage

```
public boolean createNewMessage(Message m)
```

Creates a new message to the router.

Parameters:

`m` - The message to create

Returns:

True if the creation succeeded, false if not (e.g. the message was too big for the buffer)

deleteMessage

```
public void deleteMessage(java.lang.String id,
                           boolean drop)
```

Deletes a message from the buffer and informs message listeners about the event

Parameters:

`id` - Identifier of the message to delete

`drop` - If the message is dropped (e.g. because of full buffer) this should be set to true. False value indicates e.g. remove of message because it was delivered to final destination.

sortByQueueMode

```
protected java.util.List sortByQueueMode(java.util.List list)
```

Sorts/shuffles the given list according to the current sending queue mode. The list can contain either Message or Tuple objects. Other objects cause error.

Parameters:

list - The list to sort or shuffle

Returns:

The sorted/shuffled list

compareByQueueMode

```
protected int compareByQueueMode(Message m1,  
                                Message m2)
```

Gives the order of the two given messages as defined by the current queue mode

Parameters:

m1 - The first message

m2 - The second message

Returns:

-1 if the first message should come first, 1 if the second message should come first, or 0 if the ordering isn't defined

getRoutingInfo

```
public RoutingInfo getRoutingInfo()
```

Returns routing information about this router.

Returns:

The routing information.

addApplication

```
public void addApplication(Application app)
```

Adds an application to the attached applications list.

Parameters:

app - The application to attach to this router.

getApplications

```
public java.util.Collection<Application> getApplications(java.lang.String ID)
```

Returns all the applications that want to receive messages for the given application ID.

Parameters:

ID - The application ID or null for all apps.

Returns:

A list of all applications that want to receive the message.

replicate

```
public abstract MessageRouter replicate()
```

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Returns:

The replicate

toString

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

Returns a String presentation of this router

Overrides:

toString in class `java.lang.Object`

Returns:

A String presentation of this router

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class MessageStatsReport

```
java.lang.Object
  └─ report.Report
    └─ report.MessageStatsReport
```

All Implemented Interfaces:

[MessageListener](#)

```
public class MessageStatsReport
extends Report
implements MessageListener
```

Report for generating different kind of total statistics about message relaying performance. Messages that were created during the warm up period are ignored.

Note: if some statistics could not be created (e.g. overhead ratio if no messages were delivered) "NaN" is reported for double values and zero for integer median(s).

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[MessageStatsReport\(\)](#)

Constructor.

Method Summary

void	done() Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
protected void	init() Initializes the report output.
void	messageDeleted(Message m, DTNHost where, boolean dropped) Method is called when a message is deleted
void	messageTransferAborted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer was aborted before it finished
void	messageTransferred(Message m, DTNHost from, DTNHost to, boolean finalTarget) Method is called when a message is successfully transferred from a node to another.
void	messageTransferStarted(Message m, DTNHost from, DTNHost to) Method is called when a message's transfer is started

```
void newMessage(Message m)
    Method is called when a new message is created
```

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
getSettings, getSimTime, getVariance, isWarmup, isWarmupID, newEvent, removeWarmupID,
setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

MessageStatsReport

```
public MessageStatsReport()
```

Constructor.

Method Detail

init

```
protected void init()
```

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [Report](#)

messageDeleted

```
public void messageDeleted(Message m,
                           DTNHost where,
                           boolean dropped)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is deleted

Specified by:

[messageDeleted](#) in interface [MessageListener](#)

Parameters:

m - The message that was deleted

where - The host where the message was deleted

dropped - True if the message was dropped, false if removed

messageTransferAborted

```
public void messageTransferAborted(Message m,
                                DTNHost from,
```

[DTNHost](#) to)

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer was aborted before it finished

Specified by:

[messageTransferAborted](#) in interface [MessageListener](#)

Parameters:

- m - The message that was being transferred
- from - Node where the message was being transferred from
- to - Node where the message was being transferred to

messageTransferred

```
public void messageTransferred(Message m,
                             DTNHost from,
                             DTNHost to,
                             boolean finalTarget)
```

Description copied from interface: [MessageListener](#)

Method is called when a message is successfully transferred from a node to another.

Specified by:

[messageTransferred](#) in interface [MessageListener](#)

Parameters:

- m - The message that was transferred
- from - Node where the message was transferred from
- to - Node where the message was transferred to
- finalTarget - Was the target node final destination of the message and received this message for the first time.

newMessage

```
public void newMessage(Message m)
```

Description copied from interface: [MessageListener](#)

Method is called when a new message is created

Specified by:

[newMessage](#) in interface [MessageListener](#)

Parameters:

- m - Message that was created

messageTransferStarted

```
public void messageTransferStarted(Message m,
                                   DTNHost from,
                                   DTNHost to)
```

Description copied from interface: [MessageListener](#)

Method is called when a message's transfer is started

Specified by:

[messageTransferStarted](#) in interface [MessageListener](#)

Parameters:

`m` - The message that is going to be transferred
`from` - Node where the message is transferred from
`to` - Node where the message is transferred to

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Class ModuleCommunicationBus

```
java.lang.Object
└ core.ModuleCommunicationBus
```

```
public class ModuleCommunicationBus
extends java.lang.Object
```

Intermodule communication bus. Works as a blackboard where modules can post data, subscribe to data changes and also poll for data values. This is fairly similar to Message class' property interface, but these values are shared for a node instead of message.

Constructor Summary

[ModuleCommunicationBus\(\)](#)

Constructor.

Method Summary

void	addProperty (java.lang.String key, java.lang.Object value) Adds a new property for this node.
double	getDouble (java.lang.String key, double naValue) Returns a double value from the communication bus.
int	getInt (java.lang.String key, int naValue) Returns an integer value from the communication bus.
java.lang.Object	getProperty (java.lang.String key) Returns an object that was stored using the given key.
void	subscribe (java.lang.String key, ModuleCommunicationListener module) Subscribes a module to changes of a certain value.
java.lang.String	toString ()
void	unsubscribe (java.lang.String key, ModuleCommunicationListener module) Removes a notification subscription
double	updateDouble (java.lang.String key, double delta) Changes the Double value with given key with the value delta
void	updateProperty (java.lang.String key, java.lang.Object value) Updates a value for an existing property.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

ModuleCommunicationBus

```
public ModuleCommunicationBus()
```

Constructor.

Method Detail

addProperty

```
public void addProperty(java.lang.String key,
                       java.lang.Object value)
                     throws SimError
```

Adds a new property for this node. The key can be any string but it should be such that no other class accidentally uses the same value. Note that, unless the value is immutable, it can be changed by any object that can call [getProperty\(java.lang.String\)](#).

Parameters:

- key - The key which is used to lookup the value
- value - The value to store

Throws:

- [SimError](#) - if there is already a value for the given key

getProperty

```
public java.lang.Object getProperty(java.lang.String key)
```

Returns an object that was stored using the given key. If such object is not found, null is returned.

Parameters:

- key - The key used to lookup the object

Returns:

- The stored object or null if it isn't found

updateProperty

```
public void updateProperty(java.lang.String key,
                           java.lang.Object value)
                         throws SimError
```

Updates a value for an existing property. For storing the value first time, [addProperty\(String, Object\)](#) should be used which checks for name space clashes.

Parameters:

- key - The key which is used to lookup the value
- value - The new value to store

Throws:

- [SimError](#)

updateDouble

```
public double updateDouble(java.lang.String key,
                           double delta)
                           throws SimError
```

Changes the Double value with given key with the value delta

Parameters:

key - The key of variable to update
delta - Value added to the old value

Returns:

The new value

Throws:

[SimError](#) - if the value with the given key was not a Double

getDouble

```
public double getDouble(java.lang.String key,
                      double naValue)
                     throws SimError
```

Returns a double value from the communication bus.

Parameters:

key - The key of the variable
naValue - The value to return if there is no value for the key

Returns:

The value of the key, or the naValue if they key was not found

Throws:

[SimError](#) - if the value with the given key was not a Double

getInt

```
public int getInt(java.lang.String key,
                  int naValue)
                 throws SimError
```

Returns an integer value from the communication bus.

Parameters:

key - The key of the variable
naValue - The value to return if there is no value for the key

Returns:

The value of the key, or the naValue if they key was not found

Throws:

[SimError](#) - if the value with the given key was not an Integer

subscribe

```
public void subscribe(java.lang.String key,
                      ModuleCommunicationListener module)
```

Subscribes a module to changes of a certain value.

Parameters:

key - The key of the value whose changes the module is interested of
module - The module to subscribe.

unsubscribe

```
public void unsubscribe(java.lang.String key,
```

[ModuleCommunicationListener](#) module)

Removes a notification subscription

Parameters:

key - The key for which the subscription should be removed
module - The module to whose subscription is removed

toString

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

Overrides:

toString in class java.lang.Object

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Interface ModuleCommunicationListener

All Known Implementing Classes:

[EnergyAwareRouter](#), [InterferenceLimitedInterface](#), [NetworkInterface](#), [SimpleBroadcastInterface](#)

```
public interface ModuleCommunicationListener
```

This interface should be implemented by classes that want to be notified of variable value changes in ModuleCommunicationBuses.

Method Summary

void	moduleValueChanged (java.lang.String key, java.lang.Object newValue)
------	--

This method is called whenever a variable, whose changes the module has registered to, changes.

Method Detail

moduleValueChanged

```
void moduleValueChanged(java.lang.String key,  
                      java.lang.Object newValue)
```

This method is called whenever a variable, whose changes the module has registered to, changes.

Parameters:

key - The name of the variable

newValue - New value for the variable

core

Interface MovementListener

All Known Implementing Classes:

[MovementNs2Report](#)

```
public interface MovementListener
```

Interface for classes that want to be informed about node movement.

Method Summary

void	initialLocation(DTNHost host, Coord location)
Method is called when a host receives its initial location from movement model.	
void	newDestination(DTNHost host, Coord destination, double speed)
Method is called every time a host receives a new destination from its movement model.	

Method Detail

newDestination

```
void newDestination(DTNHost host,
                    Coord destination,
                    double speed)
```

Method is called every time a host receives a new destination from its movement model.

Parameters:

- host - The host that got a new destination
- destination - Coordinates of the destination
- speed - Speed towards that destination

initialLocation

```
void initialLocation(DTNHost host,
                     Coord location)
```

Method is called when a host receives its initial location from movement model.

Parameters:

- host - The host that got the location
- location - Coordinates of the location

movement

Class MovementModel

```
java.lang.Object
└ movement.MovementModel
```

Direct Known Subclasses:

[ExtendedMovementModel](#), [ExternalMovement](#), [LinearFormation](#), [MapBasedMovement](#), [RandomWalk](#), [RandomWaypoint](#), [StationaryMovement](#)

```
public abstract class MovementModel
extends java.lang.Object
```

Superclass for all movement models. All subclasses must contain at least a constructor with one [Settings](#) parameter and also a copy-constructor. They must also implement the [replicate\(\)](#) method, which should return an instance of the movement model class with same parameters as the object (immutable fields can be shared, but mutable fields must be copied).

To make a new movement model do something useful, also at least [getInitialLocation\(\)](#) and [getPath\(\)](#) are worthwhile to override.

Field Summary

protected ModuleCommunicationBus	comBus
static double[]	DEF SPEEDS default setting for speed distribution
static double[]	DEF WAIT TIMES default setting for wait time distribution
protected double	maxSpeed
protected double	maxWaitTime
protected double	minSpeed
protected double	minWaitTime
static java.lang.String	MOVEMENT_MODEL_NS MovementModel namespace (where world size and rng seed settings are looked from ("MovementModel"))
protected static java.util.Random	rng common rng for all movement models in the simulation
static java.lang.String	RNG_SEED movement models' rng seed -setting id ("rngSeed")
static java.lang.String	SPEED

	node's speed CSV (min, max) -setting id ("speed")
static java.lang.String	WAIT TIME node's wait time CSV (min, max) -setting id ("waitTime")
static java.lang.String	WORLD SIZE world's size CSV (width, height) -setting id ("worldSize")

Constructor Summary

[MovementModel\(\)](#)

Empty constructor for testing purposes.

[MovementModel\(MovementModel mm\)](#)

Copyconstructor.

[MovementModel\(Settings settings\)](#)

Creates a new MovementModel based on a Settings object's settings.

Method Summary

protected double	generateSpeed() Generates and returns a speed value between min and max of the WAIT_TIME setting.
protected double	generateWaitTime() Generates and returns a suitable waiting time at the end of a path.
ModuleCommunicationBus	getComBus() Returns the module communication bus of this movement model (if any)
abstract Coord	getInitialLocation() Returns a new initial placement for a node
int	getMaxX() Returns the largest X coordinate value this model uses
int	getMaxY() Returns the largest Y coordinate value this model uses
abstract Path	getPath() Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
boolean	isActive() Returns true if this node is active at the moment (false if not)
double	nextPathAvailable() Returns a sim time when the next path is available.
abstract MovementModel	replicate() Creates a replicate of the movement model.
static void	reset() Resets all static fields to default values
void	setComBus(ModuleCommunicationBus comBus) Sets the module communication bus for this movement model
java.lang.String	toString() Returns simply the name of the movement model class

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

SPEED

```
public static final java.lang.String SPEED
```

node's speed CSV (min, max) -setting id ("speed")

See Also:

[Constant Field Values](#)

WAIT_TIME

```
public static final java.lang.String WAIT_TIME
```

node's wait time CSV (min, max) -setting id ("waitTime")

See Also:

[Constant Field Values](#)

DEF_SPEEDS

```
public static final double[] DEF_SPEEDS
```

default setting for speed distribution

DEF_WAIT_TIMES

```
public static final double[] DEF_WAIT_TIMES
```

default setting for wait time distribution

MOVEMENT_MODEL_NS

```
public static final java.lang.String MOVEMENT_MODEL_NS
```

MovementModel namespace (where world size and rng seed settings are looked from ("MovementModel")

See Also:

[Constant Field Values](#)

WORLD_SIZE

```
public static final java.lang.String WORLD_SIZE
```

world's size CSV (width, height) -setting id ("worldSize")

See Also:

[Constant Field Values](#)

RNG_SEED

```
public static final java.lang.String RNG_SEED
    movement models' rng seed -setting id ("rngSeed")
```

See Also:

[Constant Field Values](#)

rng

```
protected static java.util.Random rng
    common rng for all movement models in the simulation
```

minSpeed

```
protected double minSpeed
```

maxSpeed

```
protected double maxSpeed
```

minWaitTime

```
protected double minWaitTime
```

maxWaitTime

```
protected double maxWaitTime
```

comBus

```
protected ModuleCommunicationBus comBus
```

Constructor Detail

MovementModel

```
public MovementModel()
```

Empty constructor for testing purposes.

MovementModel

```
public MovementModel(Settings settings)
```

Creates a new MovementModel based on a Settings object's settings.

Parameters:

settings - The Settings object where the settings are read from

MovementModel

```
public MovementModel (MovementModel mm)
```

Copyconstructor. Creates a new MovementModel based on the given prototype.

Parameters:

mm - The MovementModel prototype to base the new object to

Method Detail

getMaxX

```
public int getMaxX()
```

Returns the largest X coordinate value this model uses

Returns:

Maximum of X coordinate values

getMaxY

```
public int getMaxY()
```

Returns the largest Y coordinate value this model uses

Returns:

Maximum of Y coordinate values

generateSpeed

```
protected double generateSpeed()
```

Generates and returns a speed value between min and max of the [WAIT_TIME](#) setting.

Returns:

A new speed between min and max values

generateWaitTime

```
protected double generateWaitTime()
```

Generates and returns a suitable waiting time at the end of a path. (i.e. random variable whose value is between min and max of the [WAIT_TIME](#) setting).

Returns:

The time as a double

getPath

```
public abstract Path getPath\(\)
```

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Returns:

A new path or null

getInitialLocation

```
public abstract Coord getInitialLocation\(\)
```

Returns a new initial placement for a node

Returns:

The initial coordinates for a node

isActive

```
public boolean isActive\(\)
```

Returns true if this node is active at the moment (false if not)

Returns:

true if this node is active (false if not)

nextPathAvailable

```
public double nextPathAvailable\(\)
```

Returns a sim time when the next path is available. This implementation returns a random time in future that is [WAIT_TIME](#) from now.

Returns:

The sim time when node should ask the next time for a path

setComBus

```
public void setComBus(ModuleCommunicationBus comBus)
```

Sets the module communication bus for this movement model

Parameters:

comBus - The communications bus to set

getComBus

```
public ModuleCommunicationBus getComBus\(\)
```

Returns the module communication bus of this movement model (if any)

Returns:

The communications bus or null if one is not set

toString

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

Returns simply the name of the movement model class

Overrides:

`toString` in class `java.lang.Object`

Returns:

the name of the movement model class

replicate

```
public abstract MovementModel replicate()
```

Creates a replicate of the movement model.

Returns:

A new movement model with the same settings as this model

reset

```
public static void reset()
```

Resets all static fields to default values

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class MovementNs2Report

```
java.lang.Object
└ report.Report
  └ report.MovementNs2Report
```

All Implemented Interfaces:

[MovementListener](#)

```
public class MovementNs2Report
extends Report
implements MovementListener
```

Movement report that generates suitable movement data for ns-2 simulator as described in <http://www.isi.edu/nsnam/ns/doc/node174.html>. This report ignores the warm up settings.

Field Summary

static java.lang.String	COORD_FORMAT formatting string for coordinate values ("%. ⁵ f")
static java.lang.String	DEF_NODE_ARRAY default value for the array name ("\$node_")
static java.lang.String	DEF_NS_CMD default value for the ns command ("\$ns_")
static double	EPSILON a value "close enough" to zero (1.0E-5).
static java.lang.String	NODE_ARR_S node array's name -setting id ("nodeArray")
static java.lang.String	NS_CMD_S ns command -setting id ("nsCmd")

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[MovementNs2Report\(\)](#)

Constructor.

Method Summary

void	initialLocation (DTNHost host, Coord location) Method is called when a host receives its initial location from movement model.
void	newDestination (DTNHost host, Coord dst, double speed)

Method is called every time a host receives a new destination from its movement model.

Methods inherited from class report.Report

```
addWarmupID, done, format, getAverage, getIntAverage, getIntMedian, getMedian,  

getScenarioName, getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent,  

removeWarmupID, setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

NODE_ARR_S

```
public static final java.lang.String NODE_ARR_S
```

node array's name -setting id ("nodeArray")

See Also:

[Constant Field Values](#)

NS_CMD_S

```
public static final java.lang.String NS_CMD_S
```

ns command -setting id ("nsCmd")

See Also:

[Constant Field Values](#)

DEF_NODE_ARRAY

```
public static final java.lang.String DEF_NODE_ARRAY
```

default value for the array name ("\$node_")

See Also:

[Constant Field Values](#)

DEF_NS_CMD

```
public static final java.lang.String DEF_NS_CMD
```

default value for the ns command ("\$ns_")

See Also:

[Constant Field Values](#)

EPSILON

```
public static final double EPSILON
```

a value "close enough" to zero (1.0E-5). Used for fixing zero values

See Also:

[Constant Field Values](#)

COORD_FORMAT

```
public static final java.lang.String COORD_FORMAT
```

formatting string for coordinate values ("%.⁵f")

See Also:

[Constant Field Values](#)

Constructor Detail

MovementNs2Report

```
public MovementNs2Report()
```

Constructor. Reads [NODE_ARR_S](#) and [NS_CMD_S](#) settings and uses those values as the name of the node array and ns command. If the values aren't present, default values of ["\\$node_"](#) and ["\\$ns_"](#) are used.

Method Detail

initialLocation

```
public void initialLocation(DTNHost host,  
                           Coord location)
```

Description copied from interface: [MovementListener](#)

Method is called when a host receives its initial location from movement model.

Specified by:

[initialLocation](#) in interface [MovementListener](#)

Parameters:

host - The host that got the location
location - Coordinates of the location

newDestination

```
public void newDestination(DTNHost host,  
                           Coord dst,  
                           double speed)
```

Description copied from interface: [MovementListener](#)

Method is called every time a host receives a new destination from its movement model.

Specified by:

[newDestination](#) in interface [MovementListener](#)

Parameters:

host - The host that got a new destination
dst - Coordinates of the destination

speed - Speed towards that destination

[Overview](#) **[Package](#)** **[Class](#)** **[Tree](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)****[PREV CLASS](#)** **[NEXT CLASS](#)**SUMMARY: NESTED | **[FIELD](#)** | **[CONSTR](#)** | **[METHOD](#)****[FRAMES](#)** **[NO FRAMES](#)** **[All Classes](#)**DETAIL: **[FIELD](#)** | **[CONSTR](#)** | **[METHOD](#)**

core

Class NetworkInterface

```
java.lang.Object
  ↴ core.NetworkInterface
```

All Implemented Interfaces:[ModuleCommunicationListener](#)**Direct Known Subclasses:**[InterferenceLimitedInterface](#), [SimpleBroadcastInterface](#)

```
public abstract class NetworkInterface
extends java.lang.Object
implements ModuleCommunicationListener
```

Network interface of a DTNHost. Takes care of connectivity among hosts.

Field Summary

<code>protected java.util.List<Connection></code>	<u>connections</u>
<code>protected DTNHost</code>	<u>host</u>
<code>protected java.lang.String</code>	<u>interfacetype</u>
<code>protected ConnectivityOptimizer</code>	<u>optimizer</u>
<code>static java.lang.String</code>	<u>RANGE_ID</u> <code>ModuleCommunicationBus</code> identifier for the "radio range" variable.
<code>static java.lang.String</code>	<u>SCAN_INTERVAL_ID</u> <code>ModuleCommunicationBus</code> identifier for the "scanning interval" variable.
<code>static java.lang.String</code>	<u>SCAN_INTERVAL_S</u> scanning interval -setting id ("scanInterval")
<code>static java.lang.String</code>	<u>SPEED_ID</u> <code>ModuleCommunicationBus</code> identifier for the "transmission speed" variable.
<code>static java.lang.String</code>	<u>TRANSMIT_RANGE_S</u> transmit range -setting id ("transmitRange")
<code>static java.lang.String</code>	<u>TRANSMIT_SPEED_S</u> transmit speed -setting id ("transmitSpeed")
<code>protected double</code>	<u>transmitRange</u>
<code>protected int</code>	<u>transmitSpeed</u>

Constructor Summary

[NetworkInterface\(\)](#)

For creating an empty class of a specific type

[NetworkInterface\(NetworkInterface ni\)](#)

copy constructor

[NetworkInterface\(Settings s\)](#)

For creating an empty class of a specific type

Method Summary

protected void	connect(Connection con, NetworkInterface anotherInterface) Connects this host to another host.
abstract void	connect(NetworkInterface anotherInterface) Connects the interface to another interface.
abstract void	createConnection(NetworkInterface anotherInterface) Creates a connection to another host.
void	destroyConnection(NetworkInterface anotherInterface) Disconnect a connection between this and another host.
protected void	disconnect(Connection con, NetworkInterface anotherInterface) Disconnects this host from another host.
protected void	ensurePositiveValue(double value, java.lang.String settingName) Makes sure that a value is positive
int	getAddress() Returns the network interface address.
java.util.List<Connection>	getConnections() Returns a list of currently connected connections
DTNHost	getHost() Returns the DTNHost of this interface
java.lang.String	getInterfaceType() For checking what interface type this interface is
Coord	getLocation() Returns the current location of the host of this interface.
double	getTransmitRange() Returns the transmit range of this network layer
int	getTransmitSpeed() Returns the transmit speed of this network layer
protected boolean	isConnected(NetworkInterface netinterface) Returns true if the given NetworkInterface is connected to this host.
boolean	isScanning() Checks if this interface is currently in the scanning mode
protected boolean	isWithinRange(NetworkInterface anotherInterface) Returns true if another interface is within radio range of this interface and this interface is also within radio range of the another interface.
void	moduleValueChanged(java.lang.String key, java.lang.Object newValue) This method is called by the ModuleCommunicationBus when/if someone changes the scanning interval, transmit speed, or range
abstract NetworkInterface	replicate()

Replication function	
static void	<u>reset()</u> Resets the static fields of the class
void	<u>setListeners</u>(java.util.List<ConnectionListener> cListeners) For setting the connectionListeners
void	<u>setHost</u>(DTNHost host) For setting the host - needed when a prototype is copied for several hosts
java.lang.String	<u>toString()</u> Returns a string representation of the object.
abstract void	<u>update()</u> Updates the state of current connections (ie tears down connections that are out of range, recalculates transmission speeds etc.).

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Field Detail

TRANSMIT_RANGE_S

public static final java.lang.String **TRANSMIT_RANGE_S**

transmit range -setting id ("transmitRange")

See Also:[Constant Field Values](#)

TRANSMIT_SPEED_S

public static final java.lang.String **TRANSMIT_SPEED_S**

transmit speed -setting id ("transmitSpeed")

See Also:[Constant Field Values](#)

SCAN_INTERVAL_S

public static final java.lang.String **SCAN_INTERVAL_S**

scanning interval -setting id ("scanInterval")

See Also:[Constant Field Values](#)

SCAN_INTERVAL_ID

public static final java.lang.String **SCAN_INTERVAL_ID**[ModuleCommunicationBus](#) identifier for the "scanning interval" variable.

See Also:[Constant Field Values](#)

RANGE_ID

```
public static final java.lang.String RANGE_ID
```

[ModuleCommunicationBus](#) identifier for the "radio range" variable. Value type: double

See Also:[Constant Field Values](#)

SPEED_ID

```
public static final java.lang.String SPEED_ID
```

[ModuleCommunicationBus](#) identifier for the "transmission speed" variable. Value type: integer

See Also:[Constant Field Values](#)

host

```
protected DTNHost host
```

interfacetype

```
protected java.lang.String interfacetype
```

connections

```
protected java.util.List<Connection> connections
```

transmitRange

```
protected double transmitRange
```

transmitSpeed

```
protected int transmitSpeed
```

optimizer

```
protected ConnectivityOptimizer optimizer
```

Constructor Detail

NetworkInterface

```
public NetworkInterface(Settings s)
```

For creating an empty class of a specific type

NetworkInterface

```
public NetworkInterface()
```

For creating an empty class of a specific type

NetworkInterface

```
public NetworkInterface(NetworkInterface ni)
```

copy constructor

Method Detail

reset

```
public static void reset()
```

Resets the static fields of the class

replicate

```
public abstract NetworkInterface replicate()
```

Replication function

setHost

```
public void setHost(DTNHost host)
```

For setting the host - needed when a prototype is copied for several hosts

Parameters:

host - The host where the network interface is

getInterfaceType

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

For checking what interface type this interface is

setListeners

```
public void setListeners(java.util.List<ConnectionListener> cListeners)
```

For setting the connectionListeners

Parameters:

cListeners - List of connection listeners

getAddress

```
public int getAddress()
```

Returns the network interface address.

Returns:

The address (integer)

getTransmitRange

```
public double getTransmitRange()
```

Returns the transmit range of this network layer

Returns:

the transmit range

getTransmitSpeed

```
public int getTransmitSpeed()
```

Returns the transmit speed of this network layer

Returns:

the transmit speed

getConnections

```
public java.util.List<Connection> getConnections()
```

Returns a list of currently connected connections

Returns:

a list of currently connected connections

isScanning

```
public boolean isScanning()
```

Checks if this interface is currently in the scanning mode

Returns:

True if the interface is scanning; false if not

connect

```
public abstract void connect(NetworkInterface anotherInterface)
```

Connects the interface to another interface. Overload this in a derived class. Check the requirements for the

connection to work in the derived class, then call connect(Connection, NetworkInterface) for the actual connection.

Parameters:

anotherInterface - The interface to connect to

connect

```
protected void connect(Connection con,  
                      NetworkInterface anotherInterface)
```

Connects this host to another host. The derived class should check that all pre-requisites for making a connection are satisfied before actually connecting.

Parameters:

con - The new connection object
anotherInterface - The interface to connect to

disconnect

```
protected void disconnect(Connection con,  
                        NetworkInterface anotherInterface)
```

Disconnects this host from another host. The derived class should make the decision whether to disconnect or not

Parameters:

con - The connection to tear down

isWithinRange

```
protected boolean isWithinRange(NetworkInterface anotherInterface)
```

Returns true if another interface is within radio range of this interface and this interface is also within radio range of the another interface.

Parameters:

anotherInterface - The another interface

Returns:

True if the interface is within range, false if not

isConnected

```
protected boolean isConnected(NetworkInterface netinterface)
```

Returns true if the given NetworkInterface is connected to this host.

Parameters:

netinterface - The other NetworkInterface to check

Returns:

True if the two hosts are connected

ensurePositiveValue

```
protected void ensurePositiveValue(double value,
                                  java.lang.String settingName)
```

Makes sure that a value is positive

Parameters:

value - Value to check
 settingName - Name of the setting (for error's message)

Throws:

[SettingsError](#) - if the value was not positive

update

```
public abstract void update()
```

Updates the state of current connections (ie tears down connections that are out of range, recalculates transmission speeds etc.).

moduleValueChanged

```
public void moduleValueChanged(java.lang.String key,
                             java.lang.Object newValue)
```

This method is called by the [ModuleCommunicationBus](#) when/if someone changes the scanning interval, transmit speed, or range

Specified by:

[moduleValueChanged](#) in interface [ModuleCommunicationListener](#)

Parameters:

key - Identifier of the changed value
 newValue - New value for the variable

createConnection

```
public abstract void createConnection(NetworkInterface anotherInterface)
```

Creates a connection to another host. This method does not do any checks on whether the other node is in range or active (cf. [connect\(NetworkInterface\)](#)).

Parameters:

anotherInterface - The interface to create the connection to

destroyConnection

```
public void destroyConnection(NetworkInterface anotherInterface)
```

Disconnect a connection between this and another host.

Parameters:

anotherInterface - The other host's network interface to disconnect from this host

getHost

```
public DTNHost getHost()
```

Returns the DTNHost of this interface

getLocation

```
public Coord getLocation()
```

Returns the current location of the host of this interface.

Returns:

The location

toString

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

Returns a string representation of the object.

Overrides:

`toString` in class `java.lang.Object`

Returns:

a string representation of the object.

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

gui

Class NodeChooser

```
java.lang.Object
  ↘ java.awt.Component
    ↘ java.awt.Container
      ↘ javax.swing.JComponent
        ↘ javax.swing.JPanel
          ↘ gui.NodeChooser
```

All Implemented Interfaces:

`java.awt.event.ActionListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible`

```
public class NodeChooser
extends javax.swing.JPanel
implements java.awt.event.ActionListener
```

Node chooser panel

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JPanel

`javax.swing.JPanel.AccessibleJPanel`

Nested classes/interfaces inherited from class javax.swing.JComponent

`javax.swing.JComponent.AccessibleJComponent`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

static int	<u>MAX_NODE_COUNT</u>
------------	---------------------------------------

the maximum number of nodes to show in the list per page

Fields inherited from class javax.swing.JComponent

`accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW`

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary`NodeChooser(java.util.List<DTNHost> nodes, DTNSimGUI gui)`**Method Summary**

void	actionPerformed (java.awt.event.ActionEvent e) Action listener method for buttons and node set chooser
------	---

Methods inherited from class javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

Methods inherited from class javax.swing.JComponent

```
addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains,
createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange,
fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY,
getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder,
getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu,
getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics,
getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier,
getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize,
getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes,
getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor,
getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect,
getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus,
isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile,
isRequestFocusEnabled, isValidateRoot, paint, paintBorder, paintChildren, paintComponent,
paintImmediately, paintImmediately, print, printAll, printBorder, printChildren,
printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent,
processMouseEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction,
registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener,
repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow,
requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible,
setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder,
setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered,
setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap,
setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque,
setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI,
setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update
```

Methods inherited from class java.awt.Container

```
add, add, add, add, addContainerListener, addImpl, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent,
getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder,
getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout,
getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent,
processEvent, remove, remove, removeAll, removeContainerListener, setComponentZOrder,
setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout,
transferFocusBackward, transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents,
dispatchEvent, enable, enableEvents, enableInputMethods, firePropertyChange,
```

```

firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground,
getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners,
getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests,
getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent,
getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit,
getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet,
isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable,
isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list,
location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move,
nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent,
processInputMethodEvent, processMouseWheelEvent, remove, removeComponentListener,
removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale,
setLocation, setLocation, setName, setSize, show, show, size, toString,
transferFocus, transferFocusUpCycle

```

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

MAX_NODE_COUNT

`public static final int MAX_NODE_COUNT`

the maximum number of nodes to show in the list per page

See Also:

[Constant Field Values](#)

Constructor Detail

NodeChooser

`public NodeChooser(java.util.List<DTNHost> nodes,
DTNSimGUI gui)`

Method Detail

actionPerformed

`public void actionPerformed(java.awt.event.ActionEvent e)`

Action listener method for buttons and node set chooser

Specified by:

`actionPerformed in interface java.awt.event.ActionListener`

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

gui.playfield

Class NodeGraphic

```
java.lang.Object
  └─gui.playfield.PlayFieldGraphic
    └─gui.playfield.NodeGraphic
```

```
public class NodeGraphic
extends PlayFieldGraphic
```

Visualization of a DTN Node

Field Summary

Fields inherited from class [gui.playfield.PlayFieldGraphic](#)

[scale](#)

Constructor Summary

[NodeGraphic\(DTNHost node\)](#)

Method Summary

void	draw(java.awt.Graphics2D g2) Draws the graphic component to the graphics context g2
static void	setDrawConnections(boolean draw) Sets whether node's connections to other nodes should be drawn
static void	setDrawCoverage(boolean draw) Sets whether radio coverage of nodes should be drawn
static void	setDrawnodeName(boolean draw) Sets whether node's name should be displayed

Methods inherited from class [gui.playfield.PlayFieldGraphic](#)

[getScale](#), [invScale](#), [scale](#), [scale](#), [setScale](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

NodeGraphic

```
public NodeGraphic(DTNHost node)
```

Method Detail

draw

```
public void draw(java.awt.Graphics2D g2)
```

Description copied from class: [PlayFieldGraphic](#)

Draws the graphic component to the graphics context g2

Specified by:

[draw](#) in class [PlayFieldGraphic](#)

Parameters:

g2 - The context to draw the graphics to

setDrawCoverage

```
public static void setDrawCoverage(boolean draw)
```

Sets whether radio coverage of nodes should be drawn

Parameters:

draw - If true, radio coverage is drawn

setDrawnodeName

```
public static void setDrawnodeName(boolean draw)
```

Sets whether node's name should be displayed

Parameters:

draw - If true, node's name is displayed

setDrawConnections

```
public static void setDrawConnections(boolean draw)
```

Sets whether node's connections to other nodes should be drawn

Parameters:

draw - If true, node's connections to other nodes is drawn

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement

Class OfficeActivityMovement

```
java.lang.Object
  └── movement.MovementModel
    └── movement.MapBasedMovement
      └── movement.OfficeActivityMovement
```

All Implemented Interfaces:

[SwitchableMovement](#)

```
public class OfficeActivityMovement
extends MapBasedMovement
implements SwitchableMovement
```

This class models movement at an office. If the node happens to be at some other location than the office, it first walks the shortest path to the office and then stays there until the end of the work day. A node has only works at one office.

Field Summary

static java.lang.String	NR_OF_OFFICES_SETTING
-------------------------	---------------------------------------

static java.lang.String	OFFICE_LOCATIONS_FILE_SETTING
-------------------------	---

static java.lang.String	OFFICE_MAX_WAIT_TIME_SETTING
-------------------------	--

static java.lang.String	OFFICE_MIN_WAIT_TIME_SETTING
-------------------------	--

static java.lang.String	OFFICE_SIZE_SETTING
-------------------------	-------------------------------------

static java.lang.String	OFFICE_WAIT_TIME_PARETO_COEFF_SETTING
-------------------------	---

static java.lang.String	WORK_DAY_LENGTH_SETTING
-------------------------	---

Fields inherited from class movement.MapBasedMovement

backAllowed , FILE_S , lastMapNode , MAP_BASE_MOVEMENT_NS , MAP_SELECT_S , maxPathLength , minPathLength , NROF_FILES_S
--

Fields inherited from class movement.MovementModel

comBus , DEF_SPEEDS , DEF_WAIT_TIMES , maxSpeed , maxWaitTime , minSpeed , minWaitTime , MOVEMENT_MODEL_NS , rng , RNG_SEED , SPEED , WAIT_TIME , WORLD_SIZE

Constructor Summary

(OfficeActivityMovement proto)

[OfficeActivityMovement](#)

Copyconstructor

[OfficeActivityMovement\(Settings settings\)](#)

OfficeActivityMovement constructor

Method Summary

protected double	generateWaitTime() Generates and returns a suitable waiting time at the end of a path.
Coord	getInitialLocation() Returns a (random) coordinate that is between two adjacent MapNodes
Coord	getLastLocation() Get the last location the getPath() of this movement model has returned
Coord	getOfficeLocation()
Path	getPath() Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
Coord	getRandomCoordinateInsideOffice()
boolean	isReady() Checks if the movement model is finished doing its task and it's time to switch to the next movement model.
MapBasedMovement	replicate() Creates a replicate of the movement model.
void	setLocation(Coord lastWaypoint) Tell the movement model what its current location is

Methods inherited from class movement.[MapBasedMovement](#)

[getMap](#), [getOkMapNodeType](#)s, [selectRandomOkNode](#)

Methods inherited from class movement.[MovementModel](#)

[generateSpeed](#), [getComBus](#), [getMaxX](#), [getMaxY](#), [isActive](#), [nextPathAvailable](#), [reset](#), [setComBus](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

WORK_DAY_LENGTH_SETTING

```
public static final java.lang.String WORK_DAY_LENGTH_SETTING
```

See Also:

[Constant Field Values](#)

NR_OF_OFFICES_SETTING

```
public static final java.lang.String NR_OF_OFFICES_SETTING
```

See Also:

[Constant Field Values](#)

OFFICE_SIZE_SETTING

```
public static final java.lang.String OFFICE_SIZE_SETTING
```

See Also:

[Constant Field Values](#)

OFFICE_WAIT_TIME_PARETO_COEFF_SETTING

```
public static final java.lang.String OFFICE_WAIT_TIME_PARETO_COEFF_SETTING
```

See Also:

[Constant Field Values](#)

OFFICE_MIN_WAIT_TIME_SETTING

```
public static final java.lang.String OFFICE_MIN_WAIT_TIME_SETTING
```

See Also:

[Constant Field Values](#)

OFFICE_MAX_WAIT_TIME_SETTING

```
public static final java.lang.String OFFICE_MAX_WAIT_TIME_SETTING
```

See Also:

[Constant Field Values](#)

OFFICE_LOCATIONS_FILE_SETTING

```
public static final java.lang.String OFFICE_LOCATIONS_FILE_SETTING
```

See Also:

[Constant Field Values](#)

Constructor Detail

OfficeActivityMovement

```
public OfficeActivityMovement(Settings settings)
```

OfficeActivityMovement constructor

Parameters:

settings -

OfficeActivityMovement

```
public OfficeActivityMovement(OfficeActivityMovement proto)
```

Copyconstructor

Parameters:

proto -

Method Detail

getRandomCoordinateInsideOffice

```
public Coord getRandomCoordinateInsideOffice()
```

getInitialLocation

```
public Coord getInitialLocation()
```

Description copied from class: [MapBasedMovement](#)

Returns a (random) coordinate that is between two adjacent MapNodes

Overrides:

[getInitialLocation](#) in class [MapBasedMovement](#)

Returns:

The initial coordinates for a node

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Overrides:

[getPath](#) in class [MapBasedMovement](#)

Returns:

A new path or null

generateWaitTime

```
protected double generateWaitTime()
```

Description copied from class: [MovementModel](#)

Generates and returns a suitable waiting time at the end of a path. (i.e. random variable whose value is between min and max of the [MovementModel.WAIT_TIME](#) setting).

Overrides:

[generateWaitTime](#) in class [MovementModel](#)

Returns:

The time as a double

replicate

```
public MapBasedMovement replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Overrides:

[replicate](#) in class [MapBasedMovement](#)

Returns:

A new movement model with the same settings as this model

getLastLocation

```
public Coord getLastLocation()
```

Description copied from interface: [SwitchableMovement](#)

Get the last location the getPath() of this movement model has returned

Specified by:

[getLastLocation](#) in interface [SwitchableMovement](#)

Overrides:

[getLastLocation](#) in class [MapBasedMovement](#)

Returns:

the last location

See Also:

[SwitchableMovement](#)

isReady

```
public boolean isReady()
```

Description copied from interface: [SwitchableMovement](#)

Checks if the movement model is finished doing its task and it's time to switch to the next movement model. The method should be called between getPath() calls.

Specified by:

[isReady](#) in interface [SwitchableMovement](#)

Overrides:

[isReady](#) in class [MapBasedMovement](#)

Returns:

true if ready

See Also:

[SwitchableMovement](#)

setLocation

```
public void setLocation(Coord lastWaypoint)
```

Description copied from interface: [SwitchableMovement](#)

Tell the movement model what its current location is

Specified by:

[setLocation](#) in interface [SwitchableMovement](#)

Overrides:

[setLocation](#) in class [MapBasedMovement](#)

See Also:

[SwitchableMovement](#)

getOfficeLocation

public [Coord](#) [getOfficeLocation\(\)](#)

Returns:

The location of the office

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class OneFromEachMessageGenerator

```
java.lang.Object
└ input.MessageEventGenerator
    └ input.OneFromEachMessageGenerator
```

All Implemented Interfaces:

[EventQueue](#)

```
public class OneFromEachMessageGenerator
extends MessageEventGenerator
```

Message creation -external events generator. Creates one message from every source node (defined with [MessageEventGenerator.HOST_RANGE_S](#)) to one of the destination nodes (defined with [MessageEventGenerator.TO_HOST_RANGE_S](#)). The message size, first messages time and the intervals between creating messages can be configured like with [MessageEventGenerator](#). End time is not respected, but messages are created until every from-node has created a message.

See Also:

[MessageEventGenerator](#)

Field Summary

Fields inherited from class input.MessageEventGenerator

```
HOST RANGE S, hostRange, idPrefix, MESSAGE ID PREFIX S, MESSAGE INTERVAL S, MESSAGE SIZE S,
MESSAGE TIME S, msgTime, nextEventsTime, rng, TO HOST RANGE S, toHostRange
```

Constructor Summary

[OneFromEachMessageGenerator\(Settings s\)](#)

Method Summary

ExternalEvent	nextEvent()
-------------------------------	-----------------------------

Returns the next message creation event

Methods inherited from class input.MessageEventGenerator

```
drawHostAddress, drawMessageSize, drawNextEventTimeDiff, drawToAddress, getID, nextEventsTime
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

Constructor Detail

OneFromEachMessageGenerator

```
public OneFromEachMessageGenerator(Settings s)
```

Method Detail

nextEvent

```
public ExternalEvent nextEvent()
```

Returns the next message creation event

Specified by:

[nextEvent](#) in interface [EventQueue](#)

Overrides:

[nextEvent](#) in class [MessageEventGenerator](#)

Returns:

The next event

See Also:

[EventQueue.nextEvent\(\)](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

input

Class OneToOneMessageGenerator

```
java.lang.Object
└ input.MessageEventGenerator
    └ input.OneToOneMessageGenerator
```

All Implemented Interfaces:

[EventQueue](#)

```
public class OneToOneMessageGenerator
extends MessageEventGenerator
```

Message creation -external events generator. Creates one message from source node/nodes (defined with [MessageEventGenerator.HOST_RANGE_S](#)) to all destination nodes (defined with [MessageEventGenerator.TO_HOST_RANGE_S](#)). The message size, first messages time and the intervals between creating messages can be configured like with [MessageEventGenerator](#). End time is not respected, but messages are created until there's a message for every destination node.

See Also:

[MessageEventGenerator](#)

Field Summary

Fields inherited from class input.MessageEventGenerator

```
HOST_RANGE_S, hostRange, idPrefix, MESSAGE_ID_PREFIX_S, MESSAGE_INTERVAL_S, MESSAGE_SIZE_S,
MESSAGE_TIME_S, msgTime, nextEventsTime, rng, TO_HOST_RANGE_S, toHostRange
```

Constructor Summary

[OneToOneMessageGenerator\(](#)[Settings](#) s)

Method Summary

ExternalEvent	nextEvent()
-------------------------------	-----------------------------

Returns the next message creation event

Methods inherited from class input.MessageEventGenerator

```
drawHostAddress, drawMessageSize, drawNextEventTimeDiff, drawToAddress, getID, nextEventsTime
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

Constructor Detail

OneToOneMessageGenerator

```
public OneToOneMessageGenerator(Settings s)
```

Method Detail

nextEvent

```
public ExternalEvent nextEvent()
```

Returns the next message creation event

Specified by:

[nextEvent](#) in interface [EventQueue](#)

Overrides:

[nextEvent](#) in class [MessageEventGenerator](#)

Returns:

The next event

See Also:

[EventQueue.nextEvent\(\)](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Class ParetoRNG

```
java.lang.Object
└ core.ParetoRNG
```

```
public class ParetoRNG
extends java.lang.Object
```

A random number generator for a Pareto distribution

Constructor Summary

[ParetoRNG](#)(java.util.Random rng, double k, double minValue, double maxValue)

Creates a new Pareto random number generator that makes use of a normal random number generator

Method Summary

double	getDouble()
	Returns a Pareto distributed double value

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ParetoRNG

```
public ParetoRNG(java.util.Random rng,
                 double k,
                 double minValue,
                 double maxValue)
```

Creates a new Pareto random number generator that makes use of a normal random number generator

Parameters:

- rng -
- k -
- minValue -
- maxValue -

Method Detail

getDouble

```
public double getDouble()
```

Returns a Pareto distributed double value

Returns:

a Pareto distributed double value

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

routing

Class PassiveRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.PassiveRouter
```

```
public class PassiveRouter
extends MessageRouter
```

Passive router that doesn't send anything unless commanded. This is useful for external event -controlled routing or dummy nodes. For implementation specifics, see MessageRouter class.

Field Summary

Fields inherited from class routing.MessageRouter

```
B_SIZE_S, DENIED_NO_SPACE, DENIED_OLD, DENIED_TTL, DENIED_UNSPECIFIED, MSG_TTL_S, msgTtl,
Q_MODE_FIFO, Q_MODE_RANDOM, RCV_OK, SEND_QUEUE_MODE_S, TRY_LATER_BUSY
```

Constructor Summary

protected	PassiveRouter (PassiveRouter r)
-----------	--

Copy-constructor.

	PassiveRouter (Settings s)
--	---

Method Summary

void	changedConnection (Connection con)
------	---

Informs the router about change in connections state.

MessageRouter	replicate ()
-------------------------------	------------------------------

Creates a replicate of this router.

void	update ()
------	---------------------------

Updates router.

Methods inherited from class routing.MessageRouter

```
addApplication, addToMessages, compareQueueMode, createNewMessage, deleteMessage,
getApplications, getBufferSize, getFreeBufferSize, getHost, getMessage, getMessageCollection,
getNrofMessages, getRoutingInfo, hasMessage, init, isDeliveredMessage, isIncomingMessage,
messageAborted, messageTransferred, putToIncomingBuffer, receiveMessage,
removeFromIncomingBuffer, removeFromMessages, requestDeliverableMessages, sendMessage,
sortByQueueMode, toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail

PassiveRouter

```
public PassiveRouter(Settings s)
```

PassiveRouter

```
protected PassiveRouter(PassiveRouter r)
```

Copy-constructor.

Parameters:

r - Router to copy the settings from.

Method Detail

update

```
public void update()
```

Description copied from class: [MessageRouter](#)

Updates router. This method should be called (at least once) on every simulation interval to update the status of transfer(s).

Overrides:

[update](#) in class [MessageRouter](#)

changedConnection

```
public void changedConnection(Connection con)
```

Description copied from class: [MessageRouter](#)

Informs the router about change in connections state.

Specified by:

[changedConnection](#) in class [MessageRouter](#)

Parameters:

con - The connection that changed

replicate

```
public MessageRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement

Class Path

```
java.lang.Object
└─movement.Path
```

```
public class Path
extends java.lang.Object
```

A Path between multiple Coordinates.

Constructor Summary

[Path\(\)](#)

Creates a path with zero speed.

[Path\(double speed\)](#)

Creates a path with constant speed

[Path\(Path path\)](#)

Copy constructor.

Method Summary

void	addWaypoint(Coord wp) Adds a new waypoint to the end of the path.
void	addWaypoint(Coord wp, double speed) Adds a new waypoint with a speed towards that waypoint
java.util.List< Coord >	getCoords() Returns a reference to the coordinates of this path
Coord	getNextWaypoint() Returns the next waypoint on this path
double	getSpeed() Returns the speed towards the next waypoint (asked with getNextWaypoint()).
java.util.List<java.lang.Double>	getSpeeds()
boolean	hasNext() Returns true if the path has more waypoints, false if not
void	setSpeed(double speed) Sets a constant speed for the whole path.
java.lang.String	toString() Returns a string presentation of the path's coordinates

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail

Path

```
public Path()
```

Creates a path with zero speed.

Path

```
public Path(Path path)
```

Copy constructor. Creates a copy of this path with a shallow copy of the coordinates and speeds.

Parameters:

path - The path to create the copy from

Path

```
public Path(double speed)
```

Creates a path with constant speed

Parameters:

speed - The speed on the path

Method Detail

setSpeed

```
public void setSpeed(double speed)
```

Sets a constant speed for the whole path. Any previously set speed(s) is discarded.

getCoords

```
public java.util.List<Coord> getCoords()
```

Returns a reference to the coordinates of this path

Returns:

coordinates of the path

addWaypoint

```
public void addWaypoint(Coord wp)
```

Adds a new waypoint to the end of the path.

Parameters:

wp

- The waypoint to add
-

addWaypoint

```
public void addWaypoint(Coord wp,  
                      double speed)
```

Adds a new waypoint with a speed towards that waypoint

Parameters:

- wp - The waypoint
 - speed - The speed towards that waypoint
-

getNextWaypoint

```
public Coord getNextWaypoint()
```

Returns the next waypoint on this path

Returns:

- the next waypoint
-

hasNext

```
public boolean hasNext()
```

Returns true if the path has more waypoints, false if not

Returns:

- true if the path has more waypoints, false if not
-

getSpeed

```
public double getSpeed()
```

Returns the speed towards the next waypoint (asked with [getNextWaypoint\(\)](#)).

Returns:

- the speed towards the next waypoint
-

toString

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

Returns a string presentation of the path's coordinates

Overrides:

`toString` in class `java.lang.Object`

Returns:

- Path as a string
-

getSpeeds

```
public java.util.List<java.lang.Double> getSpeeds()
```

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui.playfield

Class PathGraphic

```
java.lang.Object
└─gui.playfield.PlayFieldGraphic
    └─gui.playfield.PathGraphic
```

```
public class PathGraphic
extends PlayFieldGraphic
```

Visualization of a Path

Field Summary

Fields inherited from class [gui.playfield.PlayFieldGraphic](#)

[scale](#)

Constructor Summary

[PathGraphic\(\[Path\]\(#\) path\)](#)

Method Summary

void	draw (java.awt.Graphics2D g2)
------	--

Draws a line trough all path's coordinates.

Methods inherited from class [gui.playfield.PlayFieldGraphic](#)

[getScale](#), [invScale](#), [scale](#), [scale](#), [setScale](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

PathGraphic

```
public PathGraphic(Path path)
```

Method Detail

draw

```
public void draw(java.awt.Graphics2D g2)
```

Draws a line through all path's coordinates.

Specified by:

[draw](#) in class [PlayFieldGraphic](#)

Parameters:

g2 - The graphics context to draw to

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS NEXT CLASS

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

applications

Class PingApplication

```
java.lang.Object
  ↘ core.Application
    ↘ applications.PingApplication
```

```
public class PingApplication
extends Application
```

Simple ping application to demonstrate the application support. The application can be configured to send pings with a fixed interval or to only answer to pings it receives. When the application receives a ping it sends a pong message in response. The corresponding `PingAppReporter` class can be used to record information about the application behavior.

See Also:[PingAppReporter](#)

Field Summary

static java.lang.String	APP_ID Application ID
static java.lang.String	PING_DEST_RANGE Destination address range - inclusive lower, exclusive upper
static java.lang.String	PING_INTERVAL Ping generation interval
static java.lang.String	PING_OFFSET Ping interval offset - avoids synchronization of ping sending
static java.lang.String	PING_PASSIVE Run in passive mode - don't generate pings but respond
static java.lang.String	PING_PING_SIZE Size of the ping message
static java.lang.String	PING_PONG_SIZE Size of the pong message
static java.lang.String	PING_SEED Seed for the app's random number generator

Fields inherited from class core.Application

[appID](#)

Constructor Summary

[PingApplication\(PingApplication a\)](#)

Copy-constructor

[PingApplication\(Settings s\)](#)

Creates a new ping application with the given settings.

Method Summary

int	<code>getDestMax()</code>
int	<code>getDestMin()</code>
double	<code>getInterval()</code>
double	<code>getLastPing()</code>
int	<code>getPingSize()</code>
int	<code>getPongSize()</code>
int	<code>getSeed()</code>
<code>Message</code>	<code>handle(Message msg, DTNHost host)</code> Handles an incoming message.
boolean	<code>isPassive()</code>
<code>Application</code>	<code>replicate()</code>
void	<code>setDestMax(int destMax)</code>
void	<code>setDestMin(int destMin)</code>
void	<code>setInterval(double interval)</code>
void	<code>setLastPing(double lastPing)</code>
void	<code>setPassive(boolean passive)</code>
void	<code>setPingSize(int pingSize)</code>
void	<code>setPongSize(int pongSize)</code>
void	<code>setSeed(int seed)</code>
void	<code>update(DTNHost host)</code> Sends a ping packet if this is an active application instance.

Methods inherited from class core.[Application](#)

[`getAppID`](#), [`getAppListeners`](#), [`sendEventToListeners`](#), [`setAppID`](#), [`setAppListeners`](#)

Methods inherited from class java.lang.Object

[`clone`](#), [`equals`](#), [`finalize`](#), [`getClass`](#), [`hashCode`](#), [`notify`](#), [`notifyAll`](#), [`toString`](#), [`wait`](#), [`wait`](#), [`wait`](#)

Field Detail

PING_PASSIVE

```
public static final java.lang.String PING_PASSIVE
```

Run in passive mode - don't generate pings but respond

See Also:

[Constant Field Values](#)

PING_INTERVAL

```
public static final java.lang.String PING_INTERVAL
```

Ping generation interval

See Also:

[Constant Field Values](#)

PING_OFFSET

```
public static final java.lang.String PING_OFFSET
```

Ping interval offset - avoids synchronization of ping sending

See Also:

[Constant Field Values](#)

PING_DEST_RANGE

```
public static final java.lang.String PING_DEST_RANGE
```

Destination address range - inclusive lower, exclusive upper

See Also:

[Constant Field Values](#)

PING_SEED

```
public static final java.lang.String PING_SEED
```

Seed for the app's random number generator

See Also:

[Constant Field Values](#)

PING_PING_SIZE

```
public static final java.lang.String PING_PING_SIZE
```

Size of the ping message

See Also:

[Constant Field Values](#)

PING_PONG_SIZE

```
public static final java.lang.String PING_PONG_SIZE
```

Size of the pong message

See Also:

[Constant Field Values](#)

APP_ID

```
public static final java.lang.String APP_ID
```

Application ID

See Also:

[Constant Field Values](#)

Constructor Detail

PingApplication

```
public PingApplication(Settings s)
```

Creates a new ping application with the given settings.

Parameters:

s - Settings to use for initializing the application.

PingApplication

```
public PingApplication(PingApplication a)
```

Copy-constructor

Parameters:

a -

Method Detail

handle

```
public Message handle(Message msg,  
                    DTNHost host)
```

Handles an incoming message. If the message is a ping message replies with a pong message. Generates events for ping and pong messages.

Specified by:

[handle](#) in class [Application](#)

Parameters:

msg - message received by the router

host - host to which the application instance is attached

Returns:

the (possibly modified) message to forward or null if the application wants the router to stop forwarding the message.

replicate

```
public Application replicate()
```

Specified by:

[replicate](#) in class [Application](#)

update

```
public void update(DTNHost host)
```

Sends a ping packet if this is an active application instance.

Specified by:

[update](#) in class [Application](#)

Parameters:

host - to which the application instance is attached

getLastPing

```
public double getLastPing()
```

Returns:

the lastPing

setLastPing

```
public void setLastPing(double lastPing)
```

Parameters:

lastPing - the lastPing to set

getInterval

```
public double getInterval()
```

Returns:

the interval

setInterval

```
public void setInterval(double interval)
```

Parameters:

interval - the interval to set

isPassive

```
public boolean isPassive()
```

Returns:

the passive

setPassive

```
public void setPassive(boolean passive)
```

Parameters:

passive - the passive to set

getDestMin

```
public int getDestMin()
```

Returns:

the destMin

setDestMin

```
public void setDestMin(int destMin)
```

Parameters:

destMin - the destMin to set

getDestMax

```
public int getDestMax()
```

Returns:

the destMax

setDestMax

```
public void setDestMax(int destMax)
```

Parameters:

destMax - the destMax to set

getSeed

```
public int getSeed()
```

Returns:

the seed

setSeed

```
public void setSeed(int seed)
```

Parameters:

seed - the seed to set

getPongSize

```
public int getPongSize()
```

Returns:

the pongSize

setPongSize

```
public void setPongSize(int pongSize)
```

Parameters:

pongSize - the pongSize to set

getPingSize

```
public int getPingSize()
```

Returns:

the pingSize

setPingSize

```
public void setPingSize(int pingSize)
```

Parameters:

pingSize - the pingSize to set

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS NEXT CLASS

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

report

Class PingAppReporter

```
java.lang.Object
└ report.Report
  └ report.PingAppReporter
```

All Implemented Interfaces:

[ApplicationListener](#)

```
public class PingAppReporter
extends Report
implements ApplicationListener
```

Reporter for the PingApplication. Counts the number of pings and pong sent and received. Calculates success probabilities.

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[PingAppReporter\(\)](#)

Method Summary

void [done\(\)](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

```
void gotEvent(java.lang.String event, java.lang.Object params, Application app, DTNHost host)
```

Application has generated an event.

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent, removeWarmupID,
setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

Constructor Detail

PingAppReporter

```
public PingAppReporter()
```

Method Detail

gotEvent

```
public void gotEvent(java.lang.String event,
                     java.lang.Object params,
                     Application app,
                     DTNHost host)
```

Description copied from interface: [ApplicationListener](#)

Application has generated an event.

Specified by:

[gotEvent](#) in interface [ApplicationListener](#)

Parameters:

- event - Event name.
- params - Additional parameters for the event
- app - Application instance that generated the event.
- host - The host this application instance is running on.

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui.playfield

Class PlayField

```
java.lang.Object
  ↘ java.awt.Component
    ↘ java.awt.Container
      ↘ javax.swing.JComponent
        ↘ javax.swing.JPanel
          ↘ gui.playfield.PlayField
```

All Implemented Interfaces:

`java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible`

```
public class PlayField
extends javax.swing.JPanel
```

The canvas where node graphics and message visualizations are drawn.

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JPanel

`javax.swing.JPanel.Accessible JPanel`

Nested classes/interfaces inherited from class javax.swing.JComponent

`javax.swing.JComponent.Accessible JComponent`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

Fields inherited from class javax.swing.JComponent

`accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW`

Fields inherited from class java.awt.Component

`BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT`

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary[PlayField\(Word w\)](#)

Creates a playfield

Method Summary

void	addMessageTransfer(DTNHost from, DTNHost to) Adds graphics for message transfer
void	addPath(Path path) Adds a path to the overlay graphics
void	clearOverlays() Removes all overlay graphics stored to be drawn
Coord	getGraphicsPosition(Coord loc) Returns the graphical presentation location for the given world location
Coord	getWorldPosition(Coord loc) Returns a world location for a given graphical location.
void	paint(java.awt.Graphics g) Draws the play field.
void	setAutoClearOverlay(boolean clear) Enables or disables the automatic clearing of overlay graphics.
void	setMap(SimMap simMap) Sets the source for the map graphics and enables map graphics showing
void	setScale(double scale) Sets the zooming/scaling factor
void	setShowMapGraphic(boolean show) Enables/disables showing of map graphics
void	setUnderlayImage(java.awt.image.BufferedImage image, double dx, double dy, double scale, double rotation) Sets an image to show under the host graphics
void	updateField() Schedule the play field to be drawn

Methods inherited from class javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

Methods inherited from class javax.swing.JComponent

```

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains,
createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange,
fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY,
getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder,
getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu,
getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics,
getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier,
getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize,
getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes,
getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor,
getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect,
getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus,
isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile,
isRequestFocusEnabled, isValidateRoot, paintBorder, paintChildren, paintComponent,
paintImmediately, print, printAll, printBorder, printChildren,
```

```
printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent,
processMouseEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction,
registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener,
repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow,
requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible,
setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder,
setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered,
setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap,
setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque,
setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI,
setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update
```

Methods inherited from class java.awt.Container

```
add, add, add, add, add, addContainerListener, addImpl, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, getComponent,
getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder,
getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout,
getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent,
processEvent, remove, remove, removeAll, removeContainerListener, setComponentZOrder,
setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout,
transferFocusBackward, transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents,
dispatchEvent, enable, enableEvents, enableInputMethods, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground,
getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners,
getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests,
getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent,
getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit,
getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet,
isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable,
isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list,
location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move,
nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent,
processInputMethodEvent, processMouseWheelEvent, remove, removeComponentListener,
removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale,
setLocation, setLocation, setName, setSize, setSize, show, show, size, toString,
transferFocus, transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail

PlayField

```
public PlayField(World w)
```

Creates a playfield

Parameters:

w - The world that contains the actors to be drawn

Method Detail

updateField

```
public void updateField()
```

Schedule the play field to be drawn

setUnderlayImage

```
public void setUnderlayImage(java.awt.image.BufferedImage image,
                           double dx,
                           double dy,
                           double scale,
                           double rotation)
```

Sets an image to show under the host graphics

Parameters:

- image - The image to set or null to remove the image
 - dx - X offset of the image
 - dy - Y offset of the image
 - scale - Image scaling factor
 - rotation - Rotatation angle of the image (radians)
-

setScale

```
public void setScale(double scale)
```

Sets the zooming/scaling factor

Parameters:

- scale - The new scale
-

setMap

```
public void setMap(SimMap simMap)
```

Sets the source for the map graphics and enables map graphics showing

Parameters:

- simMap - The map to show
-

setShowMapGraphic

```
public void setShowMapGraphic(boolean show)
```

Enables/disables showing of map graphics

Parameters:

- show - True if the map graphics should be shown (false if not)
-

setAutoClearOverlay

```
public void setAutoClearOverlay(boolean clear)
```

Enables or disables the automatic clearing of overlay graphics. If enabled, overlay graphics are cleared every time a new graphics object is set to be drawn.

Parameters:

clear - Auto clear is enabled if this is true, disabled on false

paint

```
public void paint(java.awt.Graphics g)
```

Draws the play field. To be called by Swing framework or directly if different context than screen is desired

Overrides:

paint in class javax.swing.JComponent

Parameters:

g - The graphics context to draw the field to

clearOverlays

```
public void clearOverlays()
```

Removes all overlay graphics stored to be drawn

addMessageTransfer

```
public void addMessageTransfer(DTNHost from,  
DTNHost to)
```

Adds graphics for message transfer

Parameters:

from - Who the message was from
to - Who the message was to

addPath

```
public void addPath(Path path)
```

Adds a path to the overlay graphics

Parameters:

path - Path to add

getGraphicsPosition

```
public Coord getGraphicsPosition(Coord loc)
```

Returns the graphical presentation location for the given world location

Parameters:

loc - The location to convert

Returns:

Same location in graphics space

See Also:

[getWorldPosition\(Coord\)](#)

getWorldPosition

```
public Coord getWorldPosition(Coord loc)
```

Returns a world location for a given graphical location. Note that there might be inaccuracies because of rounding.

Parameters:

loc - The location to convert

Returns:

Same location in world space

See Also:

[getGraphicsPosition\(Coord\)](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui.playfield

Class PlayFieldGraphic

```
java.lang.Object
└─gui.playfield.PlayFieldGraphic
```

Direct Known Subclasses:

[MapGraphic](#), [MessageGraphic](#), [NodeGraphic](#), [PathGraphic](#), [ScaleReferenceGraphic](#)

```
public abstract class PlayFieldGraphic
extends java.lang.Object
```

Superclass for all graphics to be drawn on the "play field".

Field Summary

<code>protected static double scale</code>	scale Common scaling factor for all playfield graphics.
--	--

Constructor Summary

[PlayFieldGraphic\(\)](#)

Method Summary

<code>abstract void draw(java.awt.Graphics2D g2)</code>	Draws the graphic component to the graphics context g2
<code>static double getScale()</code>	Returns the currently used scaling factor
<code>static double invScale(double value)</code>	Performs an inverse of the scaling procedure with current scale.
<code>static int scale(double value)</code>	Scales the value according to current zoom level
<code>static int scale(int value)</code>	Scales the value according to current zoom level
<code>static void setScale(double newScale)</code>	Set the zooming factor of the graphics to be drawn

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Field Detail

scale

```
protected static double scale
```

Common scaling factor for all playfield graphics.

See Also:

[setScale\(double\)](#)

Constructor Detail**PlayFieldGraphic**

```
public PlayFieldGraphic()
```

Method Detail**setScale**

```
public static void setScale(double newScale)
```

Set the zooming factor of the graphics to be drawn

Parameters:

newScale - New scale

getScale

```
public static double getScale()
```

Returns the currently used scaling factor

Returns:

The scaling factor

draw

```
public abstract void draw(java.awt.Graphics2D g2)
```

Draws the graphic component to the graphics context g2

Parameters:

g2 - The context to draw the graphics to

scale

```
public static int scale(double value)
```

Scales the value according to current zoom level

Parameters:

value - Value to scale

Returns:

Scaled value bit-truncated (casted) to an integer

scale

```
public static int scale(int value)
```

Scales the value according to current zoom level

Parameters:

value - Value to scale

Returns:

Scaled value bit-truncated (casted) to an integer

invScale

```
public static double invScale(double value)
```

Performs an inverse of the scaling procedure with current scale. NOTE: invScale(scale(value)) != value because of rounding to integer at scale() -methods

Parameters:

value - The value to inverse-scale

Returns:

Inverse-scaled value

See Also:

[scale\(double\)](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement.map

Class PointsOfInterest

```
java.lang.Object
└ movement.map.PointsOfInterest
```

```
public class PointsOfInterest
extends java.lang.Object
```

Handler for points of interest data.

Field Summary

static java.lang.String	POI_FILE_S Points Of Interest file path -prefix id ("poiFile")
static java.lang.String	POI_NS Points Of Interest settings namespace ("PointsOfInterest")
static java.lang.String	POI_SELECT_S Per node group setting used for selecting POI groups and their probabilities ("pois"). Syntax: poiGroupIndex1, groupSelectionProbability1, groupIndex2, prob2, etc... Sum of probabilities must be less than or equal to one (1.0).

Constructor Summary

```
PointsOfInterest(SimMap parentMap, int[] okMapNodeTypes, Settings settings,
java.util.Random rng)
Constructor.
```

Method Summary

MapNode	selectDestination() Selects a random destination from POIs or all MapNodes.
-------------------------	--

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

POI_NS

```
public static final java.lang.String POI_NS
```

Points Of Interest settings namespace ("PointsOfInterest")

See Also:[Constant Field Values](#)

POI_FILE_S

```
public static final java.lang.String POI_FILE_S
```

Points Of Interest file path -prefix id ("poiFile")

See Also:[Constant Field Values](#)

POI_SELECT_S

```
public static final java.lang.String POI_SELECT_S
```

Per node group setting used for selecting POI groups and their probabilities ("pois").

Syntax: poiGroupIndex1, groupSelectionProbability1, groupIndex2, prob2, etc...

Sum of probabilities must be less than or equal to one (1.0). If the sum is less than one, chance of getting a random MapPoint is 1-sum.

See Also:[Constant Field Values](#)

Constructor Detail

PointsOfInterest

```
public PointsOfInterest(SimMap parentMap,
                       int[] okMapNodeTypes,
                       Settings settings,
                       java.util.Random rng)
```

Constructor.

Parameters:

parentMap - The map whose MapNodes' subset the POIs are

okMapNodeTypes - Array of map node types that are OK to visit or null if all nodes are OK

settings - The Settings object where settings are read from

rng - The random number generator to use

Method Detail

selectDestination

```
public MapNode selectDestination()
```

Selects a random destination from POIs or all MapNodes. Selecting among POI groups is done by their probabilities. If sum of their probabilities is less than 1.0 and the drawn random probability is bigger than the sum, a random MapNode is selected from the SimMap.

Returns:

A destination among POIs or all MapNodes

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class ProphetRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.ProphetRouter
```

```
public class ProphetRouter
extends ActiveRouter
```

Implementation of PRoPHET router as described in *Probabilistic routing in intermittently connected networks* by Anders Lindgren et al.

Field Summary

static java.lang.String	<u>BETA_S</u> Transitivity scaling constant (beta) -setting id ("beta").
static double	<u>DEFAULT_BETA</u> delivery predictability transitivity scaling constant default value
static double	<u>GAMMA</u> delivery predictability aging constant
static double	<u>P_INIT</u> delivery predictability initialization constant
static java.lang.String	<u>PROPHET_NS</u> Prophet router's setting namespace ("ProphetRouter")
static java.lang.String	<u>SECONDS_IN_UNIT_S</u> Number of seconds in time unit -setting id ("secondsInTimeUnit").

Fields inherited from class routing.ActiveRouter

[DELETE_DELIVERED_S](#), [deleteDelivered](#), [RESPONSE_PREFIX](#), [sendingConnections](#), [TTL_CHECK_INTERVAL](#)

Fields inherited from class routing.MessageRouter

[B_SIZE_S](#), [DENIED_NO_SPACE](#), [DENIED_OLD](#), [DENIED_TTL](#), [DENIED_UNSPECIFIED](#), [MSG_TTL_S](#), [msgTtl](#), [O_MODE_FIFO](#), [O_MODE_RANDOM](#), [RCV_OK](#), [SEND_QUEUE_MODE_S](#), [TRY_LATER_BUSY](#)

Constructor Summary

protected	<u>ProphetRouter</u> (<u>ProphetRouter</u> r) Copyconstructor.
	<u>ProphetRouter</u> (<u>Settings</u> s) Constructor.

Method Summary

void |

	<code>changedConnection(Connection con)</code> Called when a connection's state changes.
double	<code>getPredFor(DTNHost host)</code> Returns the current prediction (P) value for a host or 0 if entry for the host doesn't exist.
<code>RoutingInfo</code>	<code>getRoutingInfo()</code> Returns routing information about this router.
<code>MessageRouter</code>	<code>replicate()</code> Creates a replicate of this router.
void	<code>update()</code> Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.[ActiveRouter](#)

[`addToSendingConnections`](#), [`canStartTransfer`](#), [`checkReceiving`](#), [`createNewMessage`](#), [`dropExpiredMessages`](#), [`exchangeDeliverableMessages`](#), [`getConnections`](#), [`getMessagesForConnected`](#), [`getOldestMessage`](#), [`init`](#), [`isSending`](#), [`isTransferring`](#), [`makeRoomForMessage`](#), [`makeRoomForNewMessage`](#), [`messageTransferred`](#), [`receiveMessage`](#), [`requestDeliverableMessages`](#), [`shuffleMessages`](#), [`startTransfer`](#), [`transferAborted`](#), [`transferDone`](#), [`tryAllMessages`](#), [`tryAllMessagesToAllConnections`](#), [`tryMessagesForConnected`](#), [`tryMessagesToConnections`](#)

Methods inherited from class routing.[MessageRouter](#)

[`addApplication`](#), [`addToMessages`](#), [`compareByQueueMode`](#), [`deleteMessage`](#), [`getApplications`](#), [`getBufferSize`](#), [`getFreeBufferSize`](#), [`getHost`](#), [`getMessage`](#), [`getMessageCollection`](#), [`getNrofMessages`](#), [`hasMessage`](#), [`isDeliveredMessage`](#), [`isIncomingMessage`](#), [`messageAborted`](#), [`putToIncomingBuffer`](#), [`removeFromIncomingBuffer`](#), [`removeFromMessages`](#), [`sendMessage`](#), [`sortByQueueMode`](#), [`toString`](#)

Methods inherited from class java.lang.Object

[`clone`](#), [`equals`](#), [`finalize`](#), [`getClass`](#), [`hashCode`](#), [`notify`](#), [`notifyAll`](#), [`wait`](#), [`wait`](#), [`wait`](#)

Field Detail

P_INIT

public static final double **P_INIT**

delivery predictability initialization constant

See Also:

[Constant Field Values](#)

DEFAULT_BETA

public static final double **DEFAULT_BETA**

delivery predictability transitivity scaling constant default value

See Also:

[Constant Field Values](#)

GAMMA

public static final double **GAMMA**

delivery predictability aging constant

See Also:

[Constant Field Values](#)

PROPHET_NS

```
public static final java.lang.String PROPHET_NS
```

Prophet router's setting namespace ("ProphetRouter")

See Also:

[Constant Field Values](#)

SECONDS_IN_UNIT_S

```
public static final java.lang.String SECONDS_IN_UNIT_S
```

Number of seconds in time unit -setting id ("secondsInTimeUnit"). How many seconds one time unit is when calculating aging of delivery predictions. Should be tweaked for the scenario.

See Also:

[Constant Field Values](#)

BETA_S

```
public static final java.lang.String BETA_S
```

Transitivity scaling constant (beta) -setting id ("beta"). Default value for setting is [DEFAULT_BETA](#).

See Also:

[Constant Field Values](#)

Constructor Detail

ProphetRouter

```
public ProphetRouter(Settings s)
```

Constructor. Creates a new message router based on the settings in the given Settings object.

Parameters:

s - The settings object

ProphetRouter

```
protected ProphetRouter(ProphetRouter r)
```

Copyconstructor.

Parameters:

r - The router prototype where setting values are copied from

Method Detail

changedConnection

```
public void changedConnection(Connection con)
```

Description copied from class: [ActiveRouter](#)

Called when a connection's state changes. This version doesn't do anything but subclasses may want to override this.

Overrides:

[changedConnection](#) in class [ActiveRouter](#)

Parameters:

con - The connection that changed

getPredFor

```
public double getPredFor(DTNHost host)
```

Returns the current prediction (P) value for a host or 0 if entry for the host doesn't exist.

Parameters:

host - The host to look the P for

Returns:

the current P value

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

getRoutingInfo

```
public RoutingInfo getRoutingInfo()
```

Description copied from class: [MessageRouter](#)

Returns routing information about this router.

Overrides:

[getRoutingInfo](#) in class [MessageRouter](#)

Returns:

The routing information.

replicate

```
public MessageRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class ProphetRouterWithEstimation

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.ProphetRouterWithEstimation
```

```
public class ProphetRouterWithEstimation
extends ActiveRouter
```

Implementation of PRoPHET router as described in *Probabilistic routing in intermittently connected networks* by Anders Lindgren et al. This version tries to estimate a good value of protocol parameters from a timescale parameter given by the user, and from the encounters the node sees during simulation. Refer to Karvo and Ott, *Time Scales and Delay-Tolerant Routing Protocols* Chants, 2008

Field Summary

static java.lang.String	BETA_S Transitivity scaling constant (beta) -setting id ("beta").
static double	DEFAULT_BETA delivery predictability transitivity scaling constant default value
static double	DEFAULT_PTARGET default P target
static double	GAMMA delivery predictability aging constant
static java.lang.String	P_AVG_TARGET_S Target P_avg
static double	P_INIT delivery predictability initialization constant
static java.lang.String	PROPHET_NS Prophet router's setting namespace ("ProphetRouterWithEstimation")
static java.lang.String	TIME_SCALE_S Number of seconds in time scale.

Fields inherited from class routing.ActiveRouter

[DELETE_DELIVERED_S](#), [deleteDelivered](#), [RESPONSE_PREFIX](#), [sendingConnections](#), [TTL_CHECK_INTERVAL](#)

Fields inherited from class routing.MessageRouter

[B_SIZE_S](#), [DENIED_NO_SPACE](#), [DENIED_OLD](#), [DENIED_TTL](#), [DENIED_UNSPECIFIED](#), [MSG_TTL_S](#), [msgTtl](#), [Q_MODE_FIFO](#), [Q_MODE_RANDOM](#), [RCV_OK](#), [SEND_QUEUE_MODE_S](#), [TRY_LATER_BUSY](#)

Constructor Summary

protected [ProphetRouterWithEstimation](#)([ProphetRouterWithEstimation](#) r)

	Copyconstructor.
--	------------------

	ProphetRouterWithEstimation(Settings s)
--	---

	Constructor.
--	--------------

Method Summary

void	changedConnection(Connection con) Called when a connection's state changes.
double	getPredFor(DTNHost host) Returns the current prediction (P) value for a host or 0 if entry for the host doesn't exist.
RoutingInfo	getRoutingInfo() Returns routing information about this router.
MessageRouter	replicate() Creates a replicate of this router.
void	update() Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.[ActiveRouter](#)

[addToSendConnections](#), [canStartTransfer](#), [checkReceiving](#), [createNewMessage](#), [dropExpiredMessages](#), [exchangeDeliverableMessages](#), [getConnections](#), [getMessagesForConnected](#), [getOldestMessage](#), [init](#), [isSending](#), [isTransferring](#), [makeRoomForMessage](#), [makeRoomForNewMessage](#), [messageTransferred](#), [receiveMessage](#), [requestDeliverableMessages](#), [shuffleMessages](#), [startTransfer](#), [transferAborted](#), [transferDone](#), [tryAllMessages](#), [tryAllMessagesToAllConnections](#), [tryMessagesForConnected](#), [tryMessagesToConnections](#)

Methods inherited from class routing.[MessageRouter](#)

[addApplication](#), [addToMessages](#), [compareByQueueMode](#), [deleteMessage](#), [getApplications](#), [getBufferSize](#), [getFreeBufferSize](#), [getHost](#), [getMessage](#), [getMessageCollection](#), [getNrofMessages](#), [hasMessage](#), [isDeliveredMessage](#), [isIncomingMessage](#), [messageAborted](#), [putToIncomingBuffer](#), [removeFromIncomingBuffer](#), [removeFromMessages](#), [sendMessage](#), [sortByQueueMode](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

P_INIT

public static final double P_INIT

delivery predictability initialization constant

See Also:

[Constant Field Values](#)

DEFAULT_BETA

public static final double DEFAULT_BETA

delivery predictability transitivity scaling constant default value

See Also:[Constant Field Values](#)

GAMMA

```
public static final double GAMMA
```

delivery predictability aging constant

See Also:[Constant Field Values](#)

DEFAULT_PTARGET

```
public static final double DEFAULT_PTARGET
```

default P target

See Also:[Constant Field Values](#)

PROPHET_NS

```
public static final java.lang.String PROPHET_NS
```

Prophet router's setting namespace ("ProphetRouterWithEstimation")

See Also:[Constant Field Values](#)

TIME_SCALE_S

```
public static final java.lang.String TIME_SCALE_S
```

Number of seconds in time scale.

See Also:[Constant Field Values](#)

P_AVG_TARGET_S

```
public static final java.lang.String P_AVG_TARGET_S
```

Target P_avg

See Also:[Constant Field Values](#)

BETA_S

```
public static final java.lang.String BETA_S
```

Transitivity scaling constant (beta) -setting id ("beta"). Default value for setting is [DEFAULT_BETA](#).

See Also:[Constant Field Values](#)

Constructor Detail

ProphetRouterWithEstimation

```
public ProphetRouterWithEstimation(Settings s)
```

Constructor. Creates a new message router based on the settings in the given Settings object.

Parameters:

s - The settings object

ProphetRouterWithEstimation

```
protected ProphetRouterWithEstimation(ProphetRouterWithEstimation r)
```

Copyconstructor.

Parameters:

r - The router prototype where setting values are copied from

Method Detail

changedConnection

```
public void changedConnection(Connection con)
```

Description copied from class: [ActiveRouter](#)

Called when a connection's state changes. This version doesn't do anything but subclasses may want to override this.

Overrides:

[changedConnection](#) in class [ActiveRouter](#)

Parameters:

con - The connection that changed

getPredFor

```
public double getPredFor(DTNHost host)
```

Returns the current prediction (P) value for a host or 0 if entry for the host doesn't exist.

Parameters:

host - The host to look the P for

Returns:

the current P value

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

getRoutingInfo

```
public RoutingInfo getRoutingInfo()
```

Description copied from class: [MessageRouter](#)

Returns routing information about this router.

Overrides:

[getRoutingInfo](#) in class [MessageRouter](#)

Returns:

The routing information.

replicate

```
public MessageRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement

Class RandomWalk

```
java.lang.Object
└ movement.MovementModel
    └ movement.RandomWalk
```

All Implemented Interfaces:

[SwitchableMovement](#)

```
public class RandomWalk
extends MovementModel
implements SwitchableMovement
```

Random Walk movement model

Field Summary

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

[RandomWalk\(Settings settings\)](#)

Method Summary

Coord	getInitialLocation()	Returns a possible (random) placement for a host
Coord	getLastLocation()	Get the last location the getPath() of this movement model has returned
Path	getPath()	Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
boolean	isReady()	Checks if the movement model is finished doing its task and it's time to switch to the next movement model.
RandomWalk	replicate()	Creates a replicate of the movement model.
void	setLocation(Coord lastWaypoint)	Tell the movement model what its current location is

Methods inherited from class movement.MovementModel

generateSpeed , generateWaitTime , getComBus , getMaxX , getMaxY , isActive , nextPathAvailable , reset , setComBus , toString
--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
--

Constructor Detail

RandomWalk

```
public RandomWalk(Settings settings)
```

Method Detail

getInitialLocation

```
public Coord getInitialLocation()
```

Returns a possible (random) placement for a host

Specified by:

[getInitialLocation](#) in class [MovementModel](#)

Returns:

Random position on the map

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Specified by:

[getPath](#) in class [MovementModel](#)

Returns:

A new path or null

replicate

```
public RandomWalk replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Specified by:

[replicate](#) in class [MovementModel](#)

Returns:

A new movement model with the same settings as this model

getLastLocation

```
public Coord getLastLocation()
```

Description copied from interface: [SwitchableMovement](#)

Get the last location the getPath() of this movement model has returned

Specified by:

[getLastLocation](#) in interface [SwitchableMovement](#)

Returns:

the last location

setLocation

```
public void setLocation(Coord lastWaypoint)
```

Description copied from interface: [SwitchableMovement](#)

Tell the movement model what its current location is

Specified by:

[setLocation](#) in interface [SwitchableMovement](#)

isReady

```
public boolean isReady()
```

Description copied from interface: [SwitchableMovement](#)

Checks if the movement model is finished doing its task and it's time to switch to the next movement model. The method should be called between getPath() calls.

Specified by:

[isReady](#) in interface [SwitchableMovement](#)

Returns:

true if ready

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement

Class RandomWaypoint

```
java.lang.Object
  └── movement.MovementModel
      └── movement.RandomWaypoint
```

Direct Known Subclasses:

[ClusterMovement](#)

```
public class RandomWaypoint
extends MovementModel
```

Random waypoint movement model. Creates zig-zag paths within the simulation area.

Field Summary

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

protected	RandomWaypoint (RandomWaypoint rwp)
-----------	--

	RandomWaypoint (Settings settings)
--	---

Method Summary

Coord	getInitialLocation ()
-----------------------	---------------------------------------

Returns a possible (random) placement for a host

Path	getPath ()
----------------------	----------------------------

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).

protected Coord	randomCoord ()
------------------------------------	--------------------------------

RandomWaypoint	replicate ()
--------------------------------	------------------------------

Creates a replicate of the movement model.

Methods inherited from class movement.MovementModel

```
generateSpeed, generateWaitTime, getComBus, getMaxX, getMaxY, isActive, nextPathAvailable,
reset, setComBus, toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail

RandomWaypoint

```
public RandomWaypoint(Settings settings)
```

RandomWaypoint

```
protected RandomWaypoint(RandomWaypoint rwp)
```

Method Detail

getInitialLocation

```
public Coord getInitialLocation()
```

Returns a possible (random) placement for a host

Specified by:

[getInitialLocation](#) in class [MovementModel](#)

Returns:

Random position on the map

getPath

```
public Path getPath()
```

Description copied from class: [MovementModel](#)

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Specified by:

[getPath](#) in class [MovementModel](#)

Returns:

A new path or null

replicate

```
public RandomWaypoint replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Specified by:

[replicate](#) in class [MovementModel](#)

Returns:

A new movement model with the same settings as this model

randomCoord

protected [Coord](#) randomCoord()

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class Report

```
java.lang.Object
└ report.Report
```

Direct Known Subclasses:

[AdjacencyGraphvizReport](#), [ConnectivityDtnsim2Report](#), [ConnectivityONEReport](#),
[ContactsDuringAnICTReport](#), [ContactsPerHourReport](#), [ContactTimesReport](#), [CreatedMessagesReport](#),
[DeliveredMessagesReport](#), [DistanceDelayReport](#), [DTN2Reporter](#), [EncountersVSUniqueEncountersReport](#),
[EnergyLevelReport](#), [EventLogReport](#), [MessageDelayReport](#), [MessageDeliveryReport](#),
[MessageGraphvizReport](#), [MessageLocationReport](#), [MessageReport](#), [MessageStatsReport](#),
[MovementNs2Report](#), [PingAppReporter](#), [TotalEncountersReport](#), [UniqueEncountersReport](#)

```
public abstract class Report
extends java.lang.Object
```

Abstract superclass for all reports. All settings defined in this class can be used for all Report classes. Some reports don't implement intervalled reports ([INTERVAL_SETTING](#)) and will ignore that setting. Most of the reports implement warm up feature ([WARMUP_S](#)) but the implementations are always report specific.

Field Summary

static int	DEF_PRECISION Default precision of formatted double values
static java.lang.String	INTERVAL_SETTING The interval (simulated seconds) of creating new settings files -setting id ("interval")
static java.lang.String	INTERVALLED_FORMAT Suffix for reports that are created on n second intervals
static java.lang.String	NAN String value for values that could not be calculated
protected java.io.PrintWriter	out The print writer used to write output.
static java.lang.String	OUT_SUFFIX Suffix of report files without explicit output
static java.lang.String	OUTPUT_SETTING The output file path of the report -setting id ("output")
static java.lang.String	PRECISION_SETTING Precision of formatted double values - setting id ("precision").
static java.lang.String	REPORT_NS Name space of the settings that are common to all reports ("Report").
static java.lang.String	REPORTDIR_SETTING The default output directory of reports (can be overridden per report with OUTPUT_SETTING) -setting id ("Report.reportDir")
static java.lang.String	

[WARMUP_S](#)

Warm up period -setting id ("warmup").

protected
java.util.Set<java.lang.String>

[warmupIDs](#)

protected int

[warmupTime](#)

Constructor Summary

[Report\(\)](#)

Constructor.

Method Summary

protected void	<u>addWarmupID</u> (java.lang.String id) Adds a new ID to the warm up ID set
void	<u>done</u> () Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
protected java.lang.String	<u>format</u> (double value) Formats a double value according to current precision setting (see PRECISION_SETTING) and returns it in a string.
java.lang.String	<u>getAverage</u> (java.util.List<java.lang.Double> values) Returns the average of double values stored in a List or "NaN" for empty lists.
java.lang.String	<u>getIntAverage</u> (java.util.List<java.lang.Integer> values) Returns the average of integer values stored in a List
int	<u>getIntMedian</u> (java.util.List<java.lang.Integer> values) Returns the median of integer values stored in a List
java.lang.String	<u>getMedian</u> (java.util.List<java.lang.Double> values) Returns the median of double values stored in a List
protected java.lang.String	<u>getScenarioName</u> () Returns the name of the scenario as read from the settings
protected <u>Settings</u>	<u>getSettings</u> () Returns a Settings object initialized for the report class' name space that uses " Report " as the secondary name space.
protected double	<u>getSimTime</u> () Returns the current simulation time from the SimClock
java.lang.String	<u>getVariance</u> (java.util.List<java.lang.Double> values) Returns the variance of the values in the List.
protected void	<u>init</u> () Initializes the report output.
protected boolean	<u>isWarmup</u> () Returns true if the warm up period is still ongoing (simTime < warmup)
protected boolean	<u>isWarmupID</u> (java.lang.String id) Returns true if the given ID is in the warm up ID set
protected void	<u>newEvent</u> () This method should be called before every new (complete) event the report logs.
protected void	<u>removeWarmupID</u> (java.lang.String id) Removes a warm up ID from the warm up ID set

protected void	setPrefix (java.lang.String txt) Sets a prefix that will be inserted before every line in the report
protected void	write (java.lang.String txt) Writes a line to report using defined prefix and <u>out</u> writer.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail**REPORT_NS**public static final java.lang.String **REPORT_NS**

Name space of the settings that are common to all reports ("Report").

See Also:[Constant Field Values](#)**INTERVAL_SETTING**public static final java.lang.String **INTERVAL_SETTING**

The interval (simulated seconds) of creating new settings files -setting id ("interval")

See Also:[Constant Field Values](#)**OUTPUT_SETTING**public static final java.lang.String **OUTPUT_SETTING**

The output file path of the report -setting id ("output")

See Also:[Constant Field Values](#)**PRECISION_SETTING**public static final java.lang.String **PRECISION_SETTING**Precision of formatted double values - setting id ("precision"). Defines the amount of decimals shown in formatted double values. Default value is [4](#).**See Also:**[Constant Field Values](#)**DEF_PRECISION**public static final int **DEF_PRECISION**

Default precision of formatted double values

See Also:

[Constant Field Values](#)

REPORTDIR_SETTING

```
public static final java.lang.String REPORTDIR_SETTING
```

The default output directory of reports (can be overridden per report with [OUTPUT_SETTING](#)) -setting id ("Report.reportDir")

See Also:

[Constant Field Values](#)

WARMUP_S

```
public static final java.lang.String WARMUP_S
```

Warm up period -setting id ("warmup"). Defines how many seconds from the beginning of the simulation should not be included in the reports. Implementation of the feature is report specific, so check out the respective report classes for details. Default is 0. Must be a positive integer or 0.

See Also:

[Constant Field Values](#)

OUT_SUFFIX

```
public static final java.lang.String OUT_SUFFIX
```

Suffix of report files without explicit output

See Also:

[Constant Field Values](#)

INTERVALLED_FORMAT

```
public static final java.lang.String INTERVALLED_FORMAT
```

Suffix for reports that are created on n second intervals

See Also:

[Constant Field Values](#)

out

```
protected java.io.PrintWriter out
```

The print writer used to write output. See [write\(String\)](#)

NAN

```
public static final java.lang.String NAN
String value for values that could not be calculated
```

See Also:

[Constant Field Values](#)

warmupTime

```
protected int warmupTime
```

warmupIDs

```
protected java.util.Set<java.lang.String> warmupIDs
```

Constructor Detail**Report**

```
public Report()
```

Constructor. Looks for a className.output setting in the Settings and if such is found, uses that as the output file name. Otherwise scenarioName_classname.txt is used as the file name.

Method Detail**init**

```
protected void init()
```

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

newEvent

```
protected void newEvent()
```

This method should be called before every new (complete) event the report logs. If the report has no meaningful use for multiple reports, the call can be omitted (then only single output file will be generated)

write

```
protected void write(java.lang.String txt)
```

Writes a line to report using defined prefix and [out](#) writer.

Parameters:

txt - Line to write

See Also:

[setPrefix\(String\)](#)

format

```
protected java.lang.String format(double value)
```

Formats a double value according to current precision setting (see [PRECISION_SETTING](#)) and returns it in a string.

Parameters:

`value` - The value to format

Returns:

Formatted value in a string

setPrefix

```
protected void setPrefix(java.lang.String txt)
```

Sets a prefix that will be inserted before every line in the report

Parameters:

`txt` - Text to use as the prefix

getScenarioName

```
protected java.lang.String getScenarioName()
```

Returns the name of the scenario as read from the settings

Returns:

the name of the scenario as read from the settings

getSimTime

```
protected double getSimTime()
```

Returns the current simulation time from the SimClock

Returns:

the current simulation time from the SimClock

isWarmup

```
protected boolean isWarmup()
```

Returns true if the warm up period is still ongoing (`simTime < warmup`)

Returns:

true if the warm up period is still ongoing, false if not

addWarmupID

```
protected void addWarmupID(java.lang.String id)
```

Adds a new ID to the warm up ID set

Parameters:

id - The ID

removeWarmupID

```
protected void removeWarmupID(java.lang.String id)
```

Removes a warm up ID from the warm up ID set

Parameters:

id - The ID to remove

isWarmupID

```
protected boolean isWarmupID(java.lang.String id)
```

Returns true if the given ID is in the warm up ID set

Parameters:

id - The ID

Returns:

true if the given ID is in the warm up ID set

getSettings

```
protected Settings getSettings()
```

Returns a Settings object initialized for the report class' name space that uses "[Report](#)" as the secondary name space.

Returns:

a Settings object initialized for the report class' name space

done

```
public void done()
```

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

getAverage

```
public java.lang.String getAverage(java.util.List<java.lang.Double> values)
```

Returns the average of double values stored in a List or "NaN" for empty lists.

Parameters:

values - The list of double values

Returns:

average of double values stored in the List in a formatted String

getIntAverage

```
public java.lang.String getIntAverage(java.util.List<java.lang.Integer> values)
```

Returns the average of integer values stored in a List

Parameters:

values - The list of values

Returns:

average of integer values stored in the List or "NaN" for empty lists.

getMedian

```
public java.lang.String getMedian(java.util.List<java.lang.Double> values)
```

Returns the median of double values stored in a List

Parameters:

values - The list of double values

Returns:

median of double values stored in the List or "NaN" for empty lists.

getIntMedian

```
public int getIntMedian(java.util.List<java.lang.Integer> values)
```

Returns the median of integer values stored in a List

Parameters:

values - The list of values

Returns:

median of integer values stored in the List or 0 for empty lists.

getVariance

```
public java.lang.String getVariance(java.util.List<java.lang.Double> values)
```

Returns the variance of the values in the List.

Parameters:

values - The list of values

Returns:

The variance of the values in the list or "NaN" if the list is empty.

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class RoutingInfo

```
java.lang.Object
  ↴
  routing.RoutingInfo
```

```
public class RoutingInfo
extends java.lang.Object
```

Class for storing routing related information in a tree form for user interface(s).

Constructor Summary

[RoutingInfo\(java.lang.Object o\)](#)

Creates a routing info based on any object.

[RoutingInfo\(java.lang.String infoText\)](#)

Creates a routing info based on a text.

Method Summary

	void	addMoreInfo(RoutingInfo info)
		Adds child info object for this routing info.

	getMoreInfo()	
		Returns the child routing infos of this info.

java.lang.String	toString()	
		Returns the info text of this routing info.

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

RoutingInfo

```
public RoutingInfo(java.lang.String infoText)
```

Creates a routing info based on a text.

Parameters:

infoText - The text of the info

RoutingInfo

```
public RoutingInfo(java.lang.Object o)
```

Creates a routing info based on any object. Object's `toString()` method's output is used as the info text.

Parameters:

- o - The object this info is based on

Method Detail

addMoreInfo

```
public void addMoreInfo(RoutingInfo info)
```

Adds child info object for this routing info.

Parameters:

- info - The info object to add.

getMoreInfo

```
public java.util.List<RoutingInfo> getMoreInfo()
```

Returns the child routing infos of this info.

Returns:

The children of this info or an empty list if this info doesn't have any children.

toString

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

Returns the info text of this routing info.

Overrides:

`toString` in class `java.lang.Object`

Returns:

The info text

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui

Class RoutingInfoWindow

```
java.lang.Object
  ↘ java.awt.Component
    ↘ java.awt.Container
      ↘ java.awt.Window
        ↘ java.awt.Frame
          ↘ javax.swing.JFrame
            ↘ gui.RoutingInfoWindow
```

All Implemented Interfaces:

`java.awt.event.ActionListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants`

```
public class RoutingInfoWindow
extends javax.swing.JFrame
implements java.awt.event.ActionListener
```

A window for displaying routing information

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JFrame

`javax.swing.JFrame.AccessibleJFrame`

Nested classes/interfaces inherited from class java.awt.Frame

`java.awt.Frame.AccessibleAWTFrame`

Nested classes/interfaces inherited from class java.awt.Window

`java.awt.Window.AccessibleAWTWindow`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

Fields inherited from class javax.swing.JFrame

```
accessibleContext, EXIT_ON_CLOSE, rootPane, rootPaneCheckingEnabled
```

Fields inherited from class java.awt.Frame

```
CROSSHAIR_CURSOR, DEFAULT_CURSOR, E_RESIZE_CURSOR, HAND_CURSOR, ICONIFIED, MAXIMIZED_BOTH,
MAXIMIZED_HORIZ, MAXIMIZED_VERT, MOVE_CURSOR, N_RESIZE_CURSOR, NE_RESIZE_CURSOR, NORMAL,
NW_RESIZE_CURSOR, S_RESIZE_CURSOR, SE_RESIZE_CURSOR, SW_RESIZE_CURSOR, TEXT_CURSOR,
W_RESIZE_CURSOR, WAIT_CURSOR
```

Fields inherited from class java.awt.Component

```
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT
```

Fields inherited from interface javax.swing.WindowConstants

```
DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE
```

Fields inherited from interface java.awt.image.ImageObserver

```
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH
```

Constructor Summary

[RoutingInfoWindow](#)(DTNHost host)

Method Summary

void	actionPerformed (java.awt.event.ActionEvent e)
------	--

Methods inherited from class javax.swing.JFrame

```
addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane,
getDefaultCloseOperation, getGlassPane, getGraphics, getJMenuBar, getLayeredPane, getRootPane,
getTransferHandler, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled, paramString,
processWindowEvent, remove, repaint, setContentPane, setDefaultCloseOperation,
setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane,
setLayout, setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update
```

Methods inherited from class java.awt.Frame

```
addNotify, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds,
getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setCursor,
setExtendedState, setMaximizedBounds, setMenuBar, setResizable, setState, setTitle,
setUndecorated
```

Methods inherited from class java.awt.Window

```
addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener,
addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle,
createBufferStrategy, createBufferStrategy, dispose, getBufferStrategy,
getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys,
getGraphicsConfiguration, getIconImages, getInputContext, getListeners, getLocale,
getModalExclusionType, getMostRecentFocusOwner, getOwnedWindows, getOwner,
getOwnerlessWindows, getToolkit, getWarningString, getWindowFocusListeners,
getWindowListeners, getWindows, getWindowStateListeners, hide, isActive, isAlwaysOnTop,
isAlwaysOnTopSupported, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform,
isShowing, pack, postEvent, processEvent, processWindowFocusEvent, processWindowStateEvent,
removeWindowFocusListener, removeWindowListener, removeWindowStateListener, reshape,
setAlwaysOnTop, setBounds, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot,
setIconImages, setLocationByPlatform, setLocationRelativeTo, setMinimumSize,
setModalExclusionType, setSize, setSize, setVisible, show, toBack, toFront
```

Methods inherited from class java.awt.Container

```
add, add, add, add, add, addContainerListener, applyComponentOrientation,
```

```
areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt,
findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents, getComponentZOrder, getContainerListeners,
getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize,
getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paint, paintComponents, preferredSize, print, printComponents,
processContainerEvent, remove, removeAll, removeContainerListener, setComponentZOrder,
setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont,
transferFocusBackward, transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, contains, createImage, createImage, createVolatileImage, createVolatileImage,
disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBaseline, getBaselineResizeBehavior, getBounds,
getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor,
getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics,
getForeground, getHeight, getHierarchyBoundsListeners, getHierarchyListeners,
getIgnoreRepaint, getInputMethodListeners, getInputMethodRequests, getKeyListeners,
getLocation, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners,
getMousePosition, getMouseWheelListeners, getName, getParent, getPeer,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getTreeLock,
getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isBackgroundSet,
isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner,
isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet,
isMinimumSizeSet, isOpaque, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp, list,
list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove,
mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll,
processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent,
processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent,
processMouseWheelEvent, removeComponentListener, removeFocusListener,
removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener,
removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint,
requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resize, resize,
setBackground, setComponentOrientation, setDropTarget, setEnabled, setFocusable,
setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setLocation,
setLocation, setMaximumSize, setName, setPreferredSize, show, size, toString, transferFocus,
transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Methods inherited from interface java.awt.MenuContainer

```
getFont, postEvent
```

Constructor Detail

RoutingInfoWindow

```
public RoutingInfoWindow(DTNHost host)
```

Method Detail

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

Specified by:

actionPerformed in interface java.awt.event.ActionListener

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui.playfield

Class ScaleReferenceGraphic

```
java.lang.Object
└─gui.playfield.PlayFieldGraphic
    └─gui.playfield.ScaleReferenceGraphic
```

```
public class ScaleReferenceGraphic
extends PlayFieldGraphic
```

Reference scale bar graphic. This is the small reference scale on the upper left corner of the playfield.

Field Summary

Fields inherited from class [gui.playfield.PlayFieldGraphic](#)

[scale](#)

Constructor Summary

[ScaleReferenceGraphic\(\)](#)

Method Summary

void [draw](#)(java.awt.Graphics2D g2)

Draws the graphic component to the graphics context g2

Methods inherited from class [gui.playfield.PlayFieldGraphic](#)

[getScale](#), [invScale](#), [scale](#), [scale](#), [setScale](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

ScaleReferenceGraphic

```
public ScaleReferenceGraphic()
```

Method Detail

draw

```
public void draw(java.awt.Graphics2D g2)
```

Description copied from class: [PlayFieldGraphic](#)

Draws the graphic component to the graphics context g2

Specified by:

[draw](#) in class [PlayFieldGraphic](#)

Parameters:

g2 - The context to draw the graphics to

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

routing.schedule

Class ScheduleDijkstra

```
java.lang.Object
└ routing.schedule.ScheduleDijkstra
```

```
public class ScheduleDijkstra
extends java.lang.Object
```

Dijkstra's shortest path implementation for schedule data

Constructor Summary

[ScheduleDijkstra](#)([ScheduleOracle](#) oracle)

Constructor.

Method Summary

<code>java.util.List<ScheduleEntry></code>	<code>getShortestPath(java.lang.Integer from, java.lang.Integer to, double time)</code>
--	---

Finds and returns the fastest path between two destinations

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

ScheduleDijkstra

```
public ScheduleDijkstra(ScheduleOracle oracle)
```

Constructor.

Parameters:

`oracle` - The schedule oracle all nodes are OK

Method Detail

getShortestPath

```
public java.util.List<ScheduleEntry> getShortestPath(java.lang.Integer from,
java.lang.Integer to,
double time)
```

Finds and returns the fastest path between two destinations

Parameters:

`from` - The source of the path
`to` - The destination of the path
`time` - The time when the path starts

Returns:

a shortest path between the source and destination nodes in a list of Integers or an empty list if such path is not available

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

input

Class ScheduledUpdatesQueue

```
java.lang.Object
└─ input.ScheduledUpdatesQueue
```

All Implemented Interfaces:

[EventQueue](#)

```
public class ScheduledUpdatesQueue
extends java.lang.Object
implements EventQueue
```

Event queue where simulation objects can request an update to happen at the specified simulation time. Multiple updates at the same time are merged to a single update.

Constructor Summary

[ScheduledUpdatesQueue\(\)](#)

Constructor.

Method Summary

void	addUpdate(double simTime) Add a new update request for the given time
ExternalEvent	nextEvent() Returns the next scheduled event or event with time Double.MAX_VALUE if there aren't any.
double	nextEventsTime() Returns the next scheduled event's time or Double.MAX_VALUE if there aren't any events left
java.lang.String	toString()

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

ScheduledUpdatesQueue

```
public ScheduledUpdatesQueue()
```

Constructor. Creates an empty update queue.

Method Detail

nextEvent

```
public ExternalEvent nextEvent( )
```

Returns the next scheduled event or event with time Double.MAX_VALUE if there aren't any.

Specified by:

[nextEvent](#) in interface [EventQueue](#)

Returns:

the next scheduled event

nextEventsTime

```
public double nextEventsTime( )
```

Returns the next scheduled event's time or Double.MAX_VALUE if there aren't any events left

Specified by:

[nextEventsTime](#) in interface [EventQueue](#)

Returns:

the next scheduled event's time

addUpdate

```
public void addUpdate(double simTime)
```

Add a new update request for the given time

Parameters:

simTime - The time when the update should happen

toString

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

Overrides:

[toString](#) in class [java.lang.Object](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

routing.schedule

Class ScheduleEntry

```
java.lang.Object
└─routing.schedule.ScheduleEntry
```

All Implemented Interfaces:

java.io.Serializable

```
public class ScheduleEntry
extends java.lang.Object
implements java.io.Serializable
```

See Also:[Serialized Form](#)

Constructor Summary

[ScheduleEntry](#)(double time, int from, int via, int to, double duration)

Constructor of new schedule entry

Method Summary

double	getDelta()
double	getDestinationTime()
double	getDuration() Return the time it takes to get from source to destination
int	getFrom()
double	getTime() Returns time + delta
int	getTo()
int	getUsageCount()
int	getVia()
void	increaseUsageCount()
void	setDelta(double delta)
java.lang.String	toString()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

ScheduleEntry

```
public ScheduleEntry(double time,
                     int from,
                     int via,
                     int to,
                     double duration)
```

Constructor of new schedule entry

Parameters:

- `time` - When the journey from "from" starts
- `from` - The source
- `via` - The node that takes us there (or -1 if n/a)
- `to` - The destination
- `duration` - Time it takes from the source to destination

Method Detail

getTime

```
public double getTime()
```

Returns time + delta

Returns:

the time

getTo

```
public int getTo()
```

Returns:

the destination

getFrom

```
public int getFrom()
```

Returns:

the source

getVia

```
public int getVia()
```

Returns:

the via

getDuration

```
public double getDuration()
```

Return the time it takes to get from source to destination

Returns:

the duration

getDestinationTime

```
public double getDestinationTime()
```

getDelta

```
public double getDelta()
```

Returns:

the delta

setDelta

```
public void setDelta(double delta)
```

Parameters:

delta - the delta to set

getUsageCount

```
public int getUsageCount()
```

Returns:

the usageCount

increaseUsageCount

```
public void increaseUsageCount()
```

toString

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

Overrides:

toString in class java.lang.Object

routing.schedule

Class ScheduleOracle

```
java.lang.Object
└ routing.schedule.ScheduleOracle
```

All Implemented Interfaces:

- java.io.Serializable

```
public class ScheduleOracle
extends java.lang.Object
implements java.io.Serializable
```

See Also:

[Serialized Form](#)

Constructor Summary

ScheduleOracle()

Method Summary

void	addEntry(double start, int from, int to, double duration) Adds a new schedule entry to the oracle
void	addEntry(double start, int from, int via, int to, double duration) Adds a new schedule entry to the oracle
<code>java.util.List<ScheduleEntry></code>	getConnected(int from, double time) Returns a list of schedule entries for nodes reachable after given time from the given node
<code>java.util.List<ScheduleEntry></code>	getEntries() Returns all schedule entries

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
--

Constructor Detail

ScheduleOracle

```
public ScheduleOracle()
```

Method Detail

addEntry

```
public void addEntry(double start,
                     int from,
                     int via,
                     int to,
                     double duration)
```

Adds a new schedule entry to the oracle

Parameters:

- `start` - Start time
- `from` - Source of the connection
- `via` - The node that goes from "from" to "via" (or -1 for n/a)
- `to` - Destination of the connection
- `duration` - How long it takes to get to destination

addEntry

```
public void addEntry(double start,
                     int from,
                     int to,
                     double duration)
```

Adds a new schedule entry to the oracle

Parameters:

- `start` - Start time
- `from` - Source of the connection
- `to` - Destination of the connection
- `duration` - How long it takes to get to destination

getConnected

```
public java.util.List<ScheduleEntry> getConnected(int from,
                                         double time)
```

Returns a list of schedule entries for nodes reachable after given time from the given node

Parameters:

- `from` - The source node
- `time` - Time to start

Returns:

List of reachable nodes

getEntries

```
public java.util.List<ScheduleEntry> getEntries()
```

Returns all schedule entries

Returns:

all schedule entries

core

Class Settings

```
java.lang.Object
└ core.Settings
```

```
public class Settings
extends java.lang.Object
```

Interface for simulation settings stored in setting file(s). Settings class should be initialized before using (with [init\(String\)](#)). If Settings isn't initialized, only settings in [DEF SETTINGS FILE](#) are read. Normally, after initialization, settings in the given file can override any settings defined in the default settings file and/or define new settings.

All settings are key-value pairs. For parsing details see [Properties.getProperty\(String\)](#). Value can be a single value or comma separated list of values. With CSV values, CSV methods must be used (e.g. [getCsvInts\(String, int\)](#)). Setting value should not start and end with a bracket since those are reserved for run-specific values (see [setRunIndex\(int\)](#)). In file paths directory separator should always be forward slash ("").

Field Summary

static java.lang.String	DEF SETTINGS FILE file name of the default settings file ("default_settings.txt")
static java.lang.String	FILL DELIMITER delimiter for requested values in strings ("% %")
protected static java.util.Properties	props properties object where the setting files are read into
static java.lang.String	SETTING OUTPUT S Setting to define the file name where all read settings are written ("Settings.output").

Constructor Summary

[Settings\(\)](#)

Create a setting object without namespace.

[Settings\(java.lang.String namespace\)](#)

Creates a setting object with a namespace.

Method Summary

static void	addsettings(java.lang.String propFile) Reads another settings file and adds the key-value pairs to the current settings overriding any values that already existed with the same keys.
void	assertValidRange(int[] range, java.lang.String sname) Checks that the given integer array contains a valid range.
boolean	contains(java.lang.String name)

	Returns true if a setting with defined name (in the current namespace or secondary namespace if such is set) exists and has some value (not just white space)
java.lang.Object	<p><code>createInitializedObject</code>(java.lang.String className)</p> <p>Creates (and dynamically loads the class of) an object that initializes itself using the settings in this Settings object (given as the only parameter to the constructor).</p>
java.lang.Object	<p><code>createObject</code>(java.lang.String className)</p> <p>Creates (and dynamically loads the class of) an object using the constructor without any parameters.</p>
boolean	<p><code>getBoolean</code>(java.lang.String name)</p> <p>Returns a boolean-valued setting</p>
double[]	<p><code>getCsvDoubles</code>(java.lang.String name)</p> <p>Returns an array of CSV setting double values.</p>
double[]	<p><code>getCsvDoubles</code>(java.lang.String name, int expectedCount)</p> <p>Returns an array of CSV setting double values containing expected amount of values.</p>
int[]	<p><code>getCsvInts</code>(java.lang.String name)</p> <p>Returns an array of CSV setting integer values</p>
int[]	<p><code>getCsvInts</code>(java.lang.String name, int expectedCount)</p> <p>Returns an array of CSV setting integer values</p>
java.lang.String[]	<p><code>getCsvSetting</code>(java.lang.String name)</p> <p>Returns a CSV setting.</p>
java.lang.String[]	<p><code>getCsvSetting</code>(java.lang.String name, int expectedCount)</p> <p>Returns a CSV setting containing expected amount of values.</p>
double	<p><code>getDouble</code>(java.lang.String name)</p> <p>Returns a double-valued setting</p>
java.lang.String	<p><code>getFullPropertyName</code>(java.lang.String setting)</p> <p>Returns full (namespace prefixed) property name for a setting.</p>
int	<p><code>getInt</code>(java.lang.String name)</p> <p>Returns an integer-valued setting</p>
java.lang.String	<p><code>getNameSpace</code>()</p> <p>Returns the namespace of the settings object</p>
java.lang.String	<p><code>getSetting</code>(java.lang.String name)</p> <p>Returns a String-valued setting.</p>
static void	<p><code>init</code>(java.lang.String propFile)</p> <p>Initializes the settings all Settings objects will use.</p>
void	<p><code>restoreNameSpace</code>()</p> <p>Restores the namespace that was in use before a call to setNameSpace</p>
void	<p><code>restoreSecondaryNamespace</code>()</p> <p>Restores the secondary namespace that was in use before a call to setSecondaryNameSpace</p>
void	<p><code>setNameSpace</code>(java.lang.String namespace)</p> <p>Sets the namespace to something else than the current namespace.</p>
static void	<p><code>setRunIndex</code>(int index)</p> <p>Sets the run index for the settings (only has effect on settings with run array).</p>
void	<p><code>setSecondaryNamespace</code>(java.lang.String namespace)</p> <p>Sets a secondary namespace where a setting is searched from if it isn't found from the primary namespace.</p>
java.lang.String	<p><code>toString</code>()</p> <p>Returns a String representation of the stored settings</p>

java.lang.String	<u>valueFillString</u> (java.lang.String input) Fills a String formatted in a special way with values from Settings.
------------------	---

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

props

protected static java.util.Properties **props**

properties object where the setting files are read into

DEF_SETTINGS_FILE

public static final java.lang.String **DEF_SETTINGS_FILE**

file name of the default settings file ("default_settings.txt")

See Also:

[Constant Field Values](#)

SETTING_OUTPUT_S

public static final java.lang.String **SETTING_OUTPUT_S**

Setting to define the file name where all read settings are written ("Settings.output"). If set to an empty string, standard output is used. By default setting are not written anywhere.

See Also:

[Constant Field Values](#)

FILL_DELIMITER

public static final java.lang.String **FILL_DELIMITER**

delimiter for requested values in strings ("%%")

See Also:

[valueFillString\(String\)](#), [Constant Field Values](#)

Constructor Detail

Settings

public **Settings**(java.lang.String namespace)

Creates a setting object with a namespace. Namespace is the prefix of the all subsequent setting requests.

Parameters:

namespace - Namespace to use

Settings

```
public Settings()
```

Create a setting object without namespace. All setting requests must be prefixed with a valid namespace (e.g. "Report.nrofReports").

Method Detail

setRunIndex

```
public static void setRunIndex(int index)
```

Sets the run index for the settings (only has effect on settings with run array). A run array can be defined with syntax

```
[settingFor1stRun ; settingFor2ndRun ; SettingFor3rdRun]
```

I.e. settings are put in brackets and delimited with semicolon. First run's setting is returned when index is 0, second when index is 1 etc. If run index is bigger than run array's length, indexing wraps around in run array (i.e. return value is the value at index `runIndex % arrayLength`). To disable whole run-index-thing, set index to value smaller than zero (e.g. -1). When disabled, run-arrays are returned as normal values, including the brackets.

Parameters:

index - The run index to use for subsequent settings calls, or -1 to disable run indexing

assertValidRange

```
public void assertValidRange(int[] range,
                             java.lang.String sname)
                             throws SettingsError
```

Checks that the given integer array contains a valid range. I.e., the length of the array must be two and `first_value <= second_value`.

Parameters:

range - The range array
sname - Name of the setting (for error messages)

Throws:

[SettingsError](#) - If the given array didn't qualify as a range

setNameSpace

```
public void setNameSpace(java.lang.String namespace)
```

Sets the namespace to something else than the current namespace. This change can be reverted using [restoreNameSpace\(\)](#)

Parameters:

namespace - The new namespace

getFullPropertyName

```
public java.lang.String getFullPropertyName(java.lang.String setting)
```

Returns full (namespace prefixed) property name for a setting.

Parameters:

setting - The name of the setting

Returns:

The setting name prefixed with fully qualified name of the namespace where the requested setting would be retrieved from or null if that setting is not found from any of the current namespace(s)

getNameSpace

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

Returns the namespace of the settings object

Returns:

the namespace of the settings object

setSecondaryNamespace

```
public void setSecondaryNamespace(java.lang.String namespace)
```

Sets a secondary namespace where a setting is searched from if it isn't found from the primary namespace. Secondary namespace can be used e.g. as a "default" space where the settings are looked from if no specific setting is set. This change can be reverted using [restoreSecondaryNamespace\(\)](#)

Parameters:

namespace - The new secondary namespace or null if secondary namespace is not used (default behavior)

restoreNameSpace

```
public void restoreNameSpace()
```

Restores the namespace that was in use before a call to setNameSpace

See Also:

[setNameSpace\(String\)](#)

restoreSecondaryNamespace

```
public void restoreSecondaryNamespace()
```

Restores the secondary namespace that was in use before a call to setSecondaryNameSpace

See Also:

[setSecondaryNamespace\(String\)](#)

init

```
public static void init(java.lang.String propFile)
    throws SettingsError
```

Initializes the settings all Settings objects will use. This should be called before any setting requests.

Subsequent calls replace all old settings and then Settings contains only the new settings. The file [DEF_SETTINGS_FILE](#), if exists, is always read.

Parameters:

`propFile` - Path to the property file where additional settings are read from or null if no additional settings files are needed.

Throws:

[SettingsError](#) - If loading the settings file(s) didn't succeed

addSettings

```
public static void addSettings(java.lang.String propFile)
    throws SettingsError
```

Reads another settings file and adds the key-value pairs to the current settings overriding any values that already existed with the same keys.

Parameters:

`propFile` - Path to the property file

Throws:

[SettingsError](#) - If loading the settings file didn't succeed

See Also:

[init\(String\)](#)

contains

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

Returns true if a setting with defined name (in the current namespace or secondary namespace if such is set) exists and has some value (not just white space)

Parameters:

`name` - Name of the setting to check

Returns:

True if the setting exists, false if not

getSetting

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

Returns a String-valued setting. Setting is first looked from the namespace that is set (if any) and then from the secondary namespace (if any). All other getters use this method as their first step too (so all getters may throw SettingsError and look from both namespaces).

Parameters:

`name` - Name of the setting to get

Returns:

The contents of the setting in a String

Throws:

[SettingsError](#) - if the setting is not found from either one of the namespaces

getDouble

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

Returns a double-valued setting

Parameters:

name - Name of the setting to get

Returns:

Value of the setting as a double

getCsvSetting

```
public java.lang.String[] getCsvSetting(java.lang.String name)
```

Returns a CSV setting. Value part of the setting must be a list of comma separated values. Whitespace between values is trimmed away.

Parameters:

name - Name of the setting

Returns:

Array of values that were comma-separated

Throws:

[SettingsError](#) - if something went wrong with reading

getCsvSetting

```
public java.lang.String[] getCsvSetting(java.lang.String name,
                                         int expectedCount)
```

Returns a CSV setting containing expected amount of values. Value part of the setting must be a list of comma separated values. Whitespace between values is trimmed away.

Parameters:

name - Name of the setting

expectedCount - how many values are expected

Returns:

Array of values that were comma-separated

Throws:

[SettingsError](#) - if something went wrong with reading or didn't read the expected amount of values.

getCsvDoubles

```
public double[] getCsvDoubles(java.lang.String name,
                               int expectedCount)
```

Returns an array of CSV setting double values containing expected amount of values.

Parameters:

name - Name of the setting

expectedCount - how many values are expected

Returns:

Array of values that were comma-separated

See Also:

[getCsvSetting\(String, int\)](#)

getCsvDoubles

```
public double[] getCsvDoubles(java.lang.String name)
```

Returns an array of CSV setting double values.

Parameters:

name - Name of the setting

Returns:

Array of values that were comma-separated

See Also:

[getCsvSetting\(String\)](#)

getCsvInts

```
public int[] getCsvInts(java.lang.String name,  
                      int expectedCount)
```

Returns an array of CSV setting integer values

Parameters:

name - Name of the setting

expectedCount - how many values are expected

Returns:

Array of values that were comma-separated

See Also:

[getCsvSetting\(String, int\)](#)

getCsvInts

```
public int[] getCsvInts(java.lang.String name)
```

Returns an array of CSV setting integer values

Parameters:

name - Name of the setting

Returns:

Array of values that were comma-separated

See Also:

[getCsvSetting\(String, int\)](#)

getInt

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

Returns an integer-valued setting

Parameters:

name - Name of the setting to get

Returns:

Value of the setting as an integer

getBoolean

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

Returns a boolean-valued setting

Parameters:

`name` - Name of the setting to get

Returns:

True if the settings value was either "true" (case ignored) or "1", false is the settings value was either "false" (case ignored) or "0".

Throws:

[SettingsError](#) - if the value wasn't any recognized value

See Also:

[getSetting\(String\)](#)

createInitializedObject

```
public java.lang.Object createInitializedObject(java.lang.String className)
```

Creates (and dynamically loads the class of) an object that initializes itself using the settings in this Settings object (given as the only parameter to the constructor).

Parameters:

`className` - Name of the class of the object

Returns:

Initialized object

Throws:

[SettingsError](#) - if object couldn't be created

createObject

```
public java.lang.Object createObject(java.lang.String className)
```

Creates (and dynamically loads the class of) an object using the constructor without any parameters.

Parameters:

`className` - Name of the class of the object

Returns:

Initialized object

Throws:

[SettingsError](#) - if object couldn't be created

valueFillString

```
public java.lang.String valueFillString(java.lang.String input)
```

Fills a String formatted in a special way with values from Settings. String can contain (fully qualified) setting names surrounded by delimiters (see [FILL_DELIMITER](#)). Values for those settings are retrieved and filled in the place of place holders.

Parameters:

`input` - The input string that may contain value requests

Returns:

A string filled with requested values (or the original string if no requests were found)

Throws:

[SettingsError](#) - if such settings were not found

toString

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

Returns a String representation of the stored settings

Overrides:

`toString` in class `java.lang.Object`

Returns:

a String representation of the stored settings

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Class SettingsError

```

java.lang.Object
  ↘ java.lang.Throwable
    ↘ java.lang.Error
      ↘ java.lang.AssertionError
        ↘ core.SimError
          ↘ core.SettingsError

```

All Implemented Interfaces:

java.io.Serializable

```

public class SettingsError
extends SimError

```

Settings related error

See Also:

[Serialized Form](#)

Constructor Summary

[SettingsError](#)(java.lang.Exception e)[SettingsError](#)(java.lang.String cause)[SettingsError](#)(java.lang.String cause, java.lang.Exception e)

Method Summary

Methods inherited from class core.SimError

[getException](#)

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

SettingsError

```
public SettingsError(java.lang.String cause)
```

SettingsError

```
public SettingsError(java.lang.String cause,  
                     java.lang.Exception e)
```

SettingsError

```
public SettingsError(java.lang.Exception e)
```

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement

Class ShortestPathMapBasedMovement

```
java.lang.Object
└ movement.MovementModel
  └ movement.MapBasedMovement
    └ movement.ShortestPathMapBasedMovement
```

All Implemented Interfaces:

[SwitchableMovement](#)

```
public class ShortestPathMapBasedMovement
extends MapBasedMovement
implements SwitchableMovement
```

Map based movement model that uses Dijkstra's algorithm to find shortest paths between two random map nodes and Points Of Interest

Field Summary

Fields inherited from class movement.MapBasedMovement

```
backAllowed, FILE_S, lastMapNode, MAP_BASE_MOVEMENT_NS, MAP_SELECT_S, maxPathLength,
minPathLength, NROF_FILES_S
```

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

	shortestPathMapBasedMovement (Settings settings) Creates a new movement model based on a Settings object's settings.
protected	shortestPathMapBasedMovement (ShortestPathMapBasedMovement mbm) Copyconstructor.

Method Summary

Path	getPath() Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is).
ShortestPathMapBasedMovement	replicate() Creates a replicate of the movement model.

Methods inherited from class movement.MapBasedMovement

```
getInitialLocation, getLastLocation, getMap, getOkMapNodeTypes, isReady, selectRandomOkNode,
setLocation
```

Methods inherited from class movement.MovementModel

```
generateSpeed, generateWaitTime, getComBus, getMaxX, getMaxY, isActive, nextPathAvailable,  
reset, setComBus, toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Methods inherited from interface movement.SwitchableMovement

```
getLastLocation, isReady, setLocation
```

Constructor Detail

ShortestPathMapBasedMovement

```
public ShortestPathMapBasedMovement(Settings settings)
```

Creates a new movement model based on a Settings object's settings.

Parameters:

settings - The Settings object where the settings are read from

ShortestPathMapBasedMovement

```
protected ShortestPathMapBasedMovement(ShortestPathMapBasedMovement mbm)
```

Copyconstructor.

Parameters:

mbm - The ShortestPathMapBasedMovement prototype to base the new object to

Method Detail

getPath

```
public Path getPath()
```

Description copied from class: MovementModel

Returns a new path by this movement model or null if no new path could be constructed at the moment (node should wait where it is). A new path should not be requested before the destination of the previous path has been reached.

Overrides:

[getPath](#) in class [MapBasedMovement](#)

Returns:

A new path or null

replicate

```
public ShortestPathMapBasedMovement replicate()
```

Description copied from class: MovementModel

Creates a replicate of the movement model.

Overrides:

[replicate](#) in class [MapBasedMovement](#)

Returns:

A new movement model with the same settings as this model

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Class SimClock

```
java.lang.Object
└ core.SimClock
```

```
public class SimClock
extends java.lang.Object
```

Wall clock for checking the simulation time.

Method Summary

void	advance(double time) Advances the time by n seconds
static SimClock	getInstance() Get the instance of the class that can also change the time.
static int	getIntTime() Returns the current time rounded to the nearest integer
static double	getTime() Returns the current time (seconds since start)
static void	reset() Resets the static fields of the class
void	setTime(double time) Sets the time of the clock.
java.lang.String	toString() Returns the current simulation time in a string

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Method Detail

getInstance

```
public static SimClock getInstance()
```

Get the instance of the class that can also change the time.

Returns:

The instance of this clock

getTime

```
public static double getTime()
```

Returns the current time (seconds since start)

Returns:

Time as a double

getIntTime

```
public static int getIntTime()
```

Returns the current time rounded to the nearest integer

Returns:

Time as integer

advance

```
public void advance(double time)
```

Advances the time by n seconds

Parameters:

time - Nrof seconds to increase the time

setTime

```
public void setTime(double time)
```

Sets the time of the clock.

Parameters:

time - the time to set

toString

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

Returns the current simulation time in a string

Overrides:

toString in class java.lang.Object

Returns:

the current simulation time in a string

reset

```
public static void reset()
```

Resets the static fields of the class

core

Class SimError

```
java.lang.Object
  ↘ java.lang.Throwable
    ↘ java.lang.Error
      ↘ java.lang.AssertionError
        ↘ core.SimError
```

All Implemented Interfaces:

[java.io.Serializable](#)

Direct Known Subclasses:

[SettingsError](#)

```
public class SimError
extends java.lang.AssertionError
```

Error in the simulation

See Also:

[Serialized Form](#)

Constructor Summary

[SimError](#)(java.lang.Exception e)[SimError](#)(java.lang.String cause)[SimError](#)(java.lang.String cause, java.lang.Exception e)

Method Summary

java.lang.Exception [getException\(\)](#)

Methods inherited from class `java.lang.Throwable`

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

SimError

```
public SimError(java.lang.String cause)
```

SimError

```
public SimError(java.lang.String cause,  
                java.lang.Exception e)
```

SimError

```
public SimError(java.lang.Exception e)
```

Method Detail

getException

```
public java.lang.Exception getException()
```

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement.map

Class SimMap

```
java.lang.Object
└ movement.map.SimMap
```

All Implemented Interfaces:

- java.io.Serializable

```
public class SimMap
extends java.lang.Object
implements java.io.Serializable
```

A simulation map for node movement.

See Also:

- [Serialized Form](#)

Constructor Summary

SimMap (java.util.Map< Coord , MapNode > nodes)

Method Summary

Coord	getMaxBound() Returns the lower right corner coordinate of the map
Coord	getMinBound() Returns the upper left corner coordinate of the map
MapNode	getNodeByCoord (Coord c) Returns a MapNode at given coordinates or null if there's no MapNode in the location of the coordinate
java.util.List< MapNode >	getNodes() Returns all the map nodes in a list
Coord	getOffset() Returns the offset that has been caused by translates made to this map (does NOT take into account mirroring).
boolean	isMirrored() Returns true if this map has been mirrored after reading
void	mirror() Mirrors all map coordinates around X axis (x'=x, y'=-y).
java.lang.String	toString() Returns a String representation of the map
void	translate (double dx, double dy) Translate whole map by dx and dy

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail**SimMap**

```
public SimMap(java.util.Map<Coord, MapNode> nodes)
```

Method Detail**getNodes**

```
public java.util.List<MapNode> getNodes()
```

Returns all the map nodes in a list

Returns:

all the map nodes in a list

getNodeByCoord

```
public MapNode getNodeByCoord(Coord c)
```

Returns a MapNode at given coordinates or null if there's no MapNode in the location of the coordinate

Parameters:

c - The coordinate

Returns:

The map node in that location or null if it doesn't exist

getMinBound

```
public Coord getMinBound()
```

Returns the upper left corner coordinate of the map

Returns:

the upper left corner coordinate of the map

getMaxBound

```
public Coord getMaxBound()
```

Returns the lower right corner coordinate of the map

Returns:

the lower right corner coordinate of the map

getOffset

```
public Coord getOffset()
```

Returns the offset that has been caused by translates made to this map (does NOT take into account mirroring).

Returns:

The current offset

isMirrored

```
public boolean isMirrored()
```

Returns true if this map has been mirrored after reading

Returns:

true if this map has been mirrored after reading

See Also:

[mirror\(\)](#)

translate

```
public void translate(double dx,  
                     double dy)
```

Translate whole map by dx and dy

Parameters:

dx - The amount to translate X coordinates

dy - the amount to translate Y coordinates

mirror

```
public void mirror()
```

Mirrors all map coordinates around X axis (x'=x, y'=-y).

toString

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

Returns a String representation of the map

Overrides:

[toString](#) in class [java.lang.Object](#)

Returns:

a String representation of the map

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

gui

Class SimMenuBar

```
java.lang.Object
  ↘ java.awt.Component
    ↘ java.awt.Container
      ↘ javax.swing.JComponent
        ↘ javax.swing.JMenuBar
          ↘ gui.SimMenuBar
```

All Implemented Interfaces:

`java.awt.event.ActionListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible, javax.swing.MenuElement`

```
public class SimMenuBar
extends javax.swing.JMenuBar
implements java.awt.event.ActionListener
```

Menu bar of the simulator GUI

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JMenuBar

`javax.swing.JMenuBar.AccessibleJMenuBar`

Nested classes/interfaces inherited from class javax.swing.JComponent

`javax.swing.JComponent.AccessibleJComponent`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

<code>static java.lang.String</code>	ABOUT_TEXT GPLv3 license text for about window
<code>static java.lang.String</code>	ABOUT_TITLE title of the about window

Fields inherited from class javax.swing.JComponent

```
accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION,
WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW
```

Fields inherited from class java.awt.Component

```
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT
```

Fields inherited from interface java.awt.image.ImageObserver

```
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH
```

Constructor Summary

[SimMenuBar](#) ([PlayField](#) field)

Method Summary

void	actionPerformed (java.awt.event.ActionEvent e)
------	--

Methods inherited from class javax.swing.JMenuBar

```
add, addNotify, getAccessibleContext, getComponent, getComponentAtIndex, getComponentIndex,
getHelpMenu, getMargin, getMenu, getMenuCount, getSelectionModel, getSubElements, getUI,
getUIClassID, isBorderPainted, isSelected, menuSelectionChanged, paintBorder, paramString,
processKeyBinding, processKeyEvent, processMouseEvent, removeNotify, setBorderPainted,
setHelpMenu, setMargin, setSelected, setSelectionModel, setUI, updateUI
```

Methods inherited from class javax.swing.JComponent

```
addAncestorListener, addVetoableChangeListener, computeVisibleRect, contains, createToolTip,
disable, enable, firePropertyChange, firePropertyChange, firePropertyChange,
fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY,
getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder,
getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu,
getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics,
getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier,
getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize,
getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes,
getRootPane, getSize, getToolTipLocation, getToolTipText, getTopLevelAncestor,
getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect,
getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus,
isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile,
isRequestFocusEnabled, isValidateRoot, paint, paintChildren, paintComponent, paintImmediately,
paintImmediately, print, printAll, printBorder, printChildren, printComponent,
processComponentKeyEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent,
putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener,
removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus,
requestFocus, requestFocusInWindow, requestFocusInWindow, resetKeyboardActions, reshape,
revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls,
setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale,
setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground,
setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize,
setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled,
setToolTipText, setTransferHandler, setUI, setVerifyInputWhenFocusTarget, setVisible,
unregisterKeyboardAction, update
```

Methods inherited from class java.awt.Container

```
add, add, add, add, add, addContainerListener, addImpl, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent,
getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder,
getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout,
getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent,
processEvent, remove, remove, removeAll, removeContainerListener, setComponentZOrder,
setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout,
transferFocusBackward, transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents,
dispatchEvent, enable, enableEvents, enableInputMethods, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground,
getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners,
getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests,
getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent,
getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit,
getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet,
isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable,
isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list,
location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move,
nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent,
processInputMethodEvent, processMouseWheelEvent, remove, removeComponentListener,
removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale,
setLocation, setLocation, setName, setSize, setSize, show, show, size, toString,
transferFocus, transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Field Detail

ABOUT_TITLE

```
public static final java.lang.String ABOUT_TITLE
```

title of the about window

See Also:

[Constant Field Values](#)

ABOUT_TEXT

```
public static final java.lang.String ABOUT_TEXT
```

GPLv3 license text for about window

See Also:

[Constant Field Values](#)

Constructor Detail

SimMenuBar

```
public SimMenuBar(PlayField field)
```

Method Detail

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

Specified by:

actionPerformed in interface java.awt.event.ActionListener

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

interfaces

Class SimpleBroadcastInterface

```
java.lang.Object
└ core.NetworkInterface
  └ interfaces.SimpleBroadcastInterface
```

All Implemented Interfaces:

[ModuleCommunicationListener](#)

```
public class SimpleBroadcastInterface
extends NetworkInterface
```

A simple Network Interface that provides a constant bit-rate service, where one transmission can be on at a time.

Field Summary

Fields inherited from class core.NetworkInterface

```
connections, host, interfacetype, optimizer, RANGE_ID, SCAN_INTERVAL_ID, SCAN_INTERVAL_S,
SPEED_ID, TRANSMIT_RANGE_S, TRANSMIT_SPEED_S, transmitRange, transmitSpeed
```

Constructor Summary

[SimpleBroadcastInterface\(Settings s\)](#)

Reads the interface settings from the Settings file

[SimpleBroadcastInterface\(SimpleBroadcastInterface ni\)](#)

Copy constructor

Method Summary

void	connect(NetworkInterface anotherInterface) Tries to connect this host to another host.
void	createConnection(NetworkInterface anotherInterface) Creates a connection to another host.
NetworkInterface	replicate() Replication function
java.lang.String	toString() Returns a string representation of the object.
void	update() Updates the state of current connections (ie tears down connections that are out of range).

Methods inherited from class core.NetworkInterface

```
connect, destroyConnection, disconnect, ensurePositiveValue, getAddress, getConnections,
getHost, getInterfaceType, getLocation, getTransmitRange, getTransmitSpeed, isConnected,
isScanning, isWithinRange, moduleValueChanged, reset, setListeners, setHost
```

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

SimpleBroadcastInterface

```
public SimpleBroadcastInterface(Settings s)
```

Reads the interface settings from the Settings file

SimpleBroadcastInterface

```
public SimpleBroadcastInterface(SimpleBroadcastInterface ni)
```

Copy constructor

Parameters:

ni - the copied network interface object

Method Detail

replicate

```
public NetworkInterface replicate()
```

Description copied from class: [NetworkInterface](#)

Replication function

Specified by:

[replicate](#) in class [NetworkInterface](#)

connect

```
public void connect(NetworkInterface anotherInterface)
```

Tries to connect this host to another host. The other host must be active and within range of this host for the connection to succeed.

Specified by:

[connect](#) in class [NetworkInterface](#)

Parameters:

anotherInterface - The interface to connect to

update

```
public void update()
```

Updates the state of current connections (ie tears down connections that are out of range).

Specified by:

[update](#) in class [NetworkInterface](#)

createConnection

```
public void createConnection(NetworkInterface anotherInterface)
```

Creates a connection to another host. This method does not do any checks on whether the other node is in range or active

Specified by:

[createConnection](#) in class [NetworkInterface](#)

Parameters:

anotherInterface - The interface to create the connection to

toString

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

Returns a string representation of the object.

Overrides:

[toString](#) in class [NetworkInterface](#)

Returns:

a string representation of the object.

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Class SimScenario

```
java.lang.Object
└ core.SimScenario
```

All Implemented Interfaces:

- java.io.Serializable

```
public class SimScenario
extends java.lang.Object
implements java.io.Serializable
```

A simulation scenario used for getting and storing the settings of a simulation run.

See Also:

- [Serialized Form](#)

Field Summary

static java.lang.String	APPCOUNT_S setting name for the number of applications
static java.lang.String	APPTYPE_NS namespace for application type settings ("Application")
static java.lang.String	APPTYPE_S application type -setting id ("type")
static java.lang.String	END_TIME_S end time -setting id ("endTime")
static java.lang.String	GAPPNAME_S application name in the group -setting id ("application")
static java.lang.String	GROUP_ID_S group id -setting id ("groupID")
static java.lang.String	GROUP_NS namespace for host group settings ("Group")
protected java.util.List< DTNHost >	hosts List of hosts in this simulation
static java.lang.String	INTERFACENAME_S interface name in the group -setting id ("interface")
static java.lang.String	INTNAME_S interface name -setting id ("name")
static java.lang.String	INTTYPE_NS namespace for interface type settings ("Interface")
static java.lang.String	INTTYPE_S interface type -setting id ("type")
static java.lang.String	MOVEMENT_MODEL_S

	movement model class -setting id ("movementModel")
static java.lang.String	NAME_S scenario name -setting id ("name")
static java.lang.String	NROF_GROUPS_S number of host groups -setting id ("nrofHostGroups")
static java.lang.String	NROF_HOSTS_S number of hosts in the group -setting id ("nrofHosts")
static java.lang.String	NROF_INTERF_S number of interfaces in the group -setting id ("nrofInterfaces")
static java.lang.String	NROF_INTTYPES_S number of interface types -setting id ("nrofInterfaceTypes")
static java.lang.String	ROUTER_S router class -setting id ("router")
static java.lang.String	SCAN_INTERVAL_S scanning interval -setting id ("scanInterval")
static java.lang.String	SCENARIO_NS namespace of scenario settings ("Scenario")
static java.lang.String	SIM_CON_S simulate connections -setting id ("simulateConnections")
static java.lang.String	UP_INT_S update interval -setting id ("updateInterval")

Constructor Summary

protected	simScenario()
	Creates a scenario based on Settings object.

Method Summary

	void addApplicationListener(ApplicationListener al) Adds a new application event listener for all nodes.
	void addConnectionListener(ConnectionListener cl) Adds a new connection listener for all nodes
	void addMessageListener(MessageListener ml) Adds a new message listener for all nodes
	void addMovementListener(MovementListener ml) Adds a new movement listener for all nodes
	void addUpdateListener(UpdateListener ul) Adds a new update listener for the world
protected void	createHosts() Creates hosts for the scenario
java.util.List<ApplicationListener>	getApplicationListeners() Returns the list of registered application event listeners
	double getEndTime() Returns simulation's end time
java.util.List<EventQueue>	getExternalEvents() Returns the (external) event queue(s) of this scenario or null if there aren't any

<code>java.util.List<DTNHost></code>	<code>getHosts()</code>	Returns the list of nodes for this scenario.
<code>static SimScenario</code>	<code>getInstance()</code>	Returns the SimScenario instance and creates one if it doesn't exist yet
<code>SimMap</code>	<code>getMap()</code>	Returns the SimMap this scenario uses, or null if scenario doesn't use any map
<code>double</code>	<code>getMaxHostRange()</code>	Returns how long range the hosts' radios have
<code>java.lang.String</code>	<code>getName()</code>	Returns the name of the simulation run
<code>double</code>	<code>getUpdateInterval()</code>	Returns update interval (simulated seconds) of the simulation
<code>java.util.List<UpdateListener></code>	<code>getUpdateListeners()</code>	Returns the list of registered update listeners
<code>World</code>	<code>getWorld()</code>	Returns the World object of this scenario
<code>int</code>	<code>getWorldSizeX()</code>	Returns the width of the world
<code>int</code>	<code>getWorldSizeY()</code>	Returns the height of the world
<code>static void</code>	<code>reset()</code>	
<code>boolean</code>	<code>simulateConnections()</code>	Returns true if connections should be simulated

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

Field Detail

SCENARIO_NS

`public static final java.lang.String SCENARIO_NS`

namespace of scenario settings ("Scenario")

See Also:

[Constant Field Values](#)

NROF_GROUPS_S

`public static final java.lang.String NROF_GROUPS_S`

number of host groups -setting id ("nrofHostGroups")

See Also:

[Constant Field Values](#)

NROF_INTTYPES_S

```
public static final java.lang.String NROF_INTTYPES_S  
    number of interface types -setting id ("nrofInterfaceTypes")
```

See Also:

[Constant Field Values](#)

NAME_S

```
public static final java.lang.String NAME_S  
    scenario name -setting id ("name")
```

See Also:

[Constant Field Values](#)

END_TIME_S

```
public static final java.lang.String END_TIME_S  
    end time -setting id ("endTime")
```

See Also:

[Constant Field Values](#)

UP_INT_S

```
public static final java.lang.String UP_INT_S  
    update interval -setting id ("updateInterval")
```

See Also:

[Constant Field Values](#)

SIM_CON_S

```
public static final java.lang.String SIM_CON_S  
    simulate connections -setting id ("simulateConnections")
```

See Also:

[Constant Field Values](#)

INTTYPE_NS

```
public static final java.lang.String INTTYPE_NS  
    namespace for interface type settings ("Interface")
```

See Also:

[Constant Field Values](#)**INTTYPE_S**

```
public static final java.lang.String INTTYPE_S  
interface type -setting id ("type")
```

See Also:[Constant Field Values](#)**INTNAME_S**

```
public static final java.lang.String INTNAME_S  
interface name -setting id ("name")
```

See Also:[Constant Field Values](#)**APPTYPE_NS**

```
public static final java.lang.String APPTYPE_NS  
namespace for application type settings ("Application")
```

See Also:[Constant Field Values](#)**APPTYPE_S**

```
public static final java.lang.String APPTYPE_S  
application type -setting id ("type")
```

See Also:[Constant Field Values](#)**APPCOUNT_S**

```
public static final java.lang.String APPCOUNT_S  
setting name for the number of applications
```

See Also:[Constant Field Values](#)**GROUP_NS**

```
public static final java.lang.String GROUP_NS  
namespace for host group settings ("Group")
```

See Also:[Constant Field Values](#)**GROUP_ID_S**

```
public static final java.lang.String GROUP_ID_S
```

group id -setting id ("groupID")

See Also:[Constant Field Values](#)**NROF_HOSTS_S**

```
public static final java.lang.String NROF_HOSTS_S
```

number of hosts in the group -setting id ("nrofHosts")

See Also:[Constant Field Values](#)**SCAN_INTERVAL_S**

```
public static final java.lang.String SCAN_INTERVAL_S
```

scanning interval -setting id ("scanInterval")

See Also:[Constant Field Values](#)**MOVEMENT_MODEL_S**

```
public static final java.lang.String MOVEMENT_MODEL_S
```

movement model class -setting id ("movementModel")

See Also:[Constant Field Values](#)**ROUTER_S**

```
public static final java.lang.String ROUTER_S
```

router class -setting id ("router")

See Also:[Constant Field Values](#)**NROF_INTERF_S**

```
public static final java.lang.String NROF_INTERF_S
```

number of interfaces in the group -setting id ("nrofInterfaces")

See Also:[Constant Field Values](#)**INTERFACENAME_S**

```
public static final java.lang.String INTERFACENAME_S
```

interface name in the group -setting id ("interface")

See Also:[Constant Field Values](#)**GAPPNAME_S**

```
public static final java.lang.String GAPPNAME_S
```

application name in the group -setting id ("application")

See Also:[Constant Field Values](#)**hosts**

```
protected java.util.List<DTNHost> hosts
```

List of hosts in this simulation

Constructor Detail**SimScenario**

```
protected SimScenario()
```

Creates a scenario based on Settings object.

Method Detail**reset**

```
public static void reset()
```

getInstance

```
public static SimScenario getInstance()
```

Returns the SimScenario instance and creates one if it doesn't exist yet

getName

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

Returns the name of the simulation run

Returns:

the name of the simulation run

simulateConnections

```
public boolean simulateConnections()
```

Returns true if connections should be simulated

Returns:

true if connections should be simulated (false if not)

getWorldSizeX

```
public int getWorldSizeX()
```

Returns the width of the world

Returns:

the width of the world

getWorldSizeY

```
public int getWorldSizeY()
```

Returns the height of the world

Returns:

the height of the world

getEndTime

```
public double getEndTime()
```

Returns simulation's end time

Returns:

simulation's end time

getUpdateInterval

```
public double getUpdateInterval()
```

Returns update interval (simulated seconds) of the simulation

Returns:

update interval (simulated seconds) of the simulation

getMaxHostRange

```
public double getMaxHostRange()
```

Returns how long range the hosts' radios have

Returns:

Range in meters

getExternalEvents

```
public java.util.List<EventQueue> getExternalEvents()
```

Returns the (external) event queue(s) of this scenario or null if there aren't any

Returns:

External event queues in a list or null

getMap

```
public SimMap getMap()
```

Returns the SimMap this scenario uses, or null if scenario doesn't use any map

Returns:

SimMap or null if no map is used

addConnectionListener

```
public void addConnectionListener(ConnectionListener cl)
```

Adds a new connection listener for all nodes

Parameters:

cl - The listener

addMessageListener

```
public void addMessageListener(MessageListener ml)
```

Adds a new message listener for all nodes

Parameters:

ml - The listener

addMovementListener

```
public void addMovementListener(MovementListener ml)
```

Adds a new movement listener for all nodes

Parameters:

ml - The listener

addUpdateListener

```
public void addUpdateListener(UpdateListener ul)
```

Adds a new update listener for the world

Parameters:

ul - The listener

getUpdateListeners

```
public java.util.List<UpdateListener> getUpdateListeners()
```

Returns the list of registered update listeners

Returns:

the list of registered update listeners

addApplicationListener

```
public void addApplicationListener(ApplicationListener al)
```

Adds a new application event listener for all nodes.

Parameters:

al - The listener

getApplicationListeners

```
public java.util.List<ApplicationListener> getApplicationListeners()
```

Returns the list of registered application event listeners

Returns:

the list of registered application event listeners

createHosts

```
protected void createHosts()
```

Creates hosts for the scenario

getHosts

```
public java.util.List<DTNHost> getHosts()
```

Returns the list of nodes for this scenario.

Returns:

the list of nodes for this scenario.

getWorld

```
public World getWorld()
```

Returns the World object of this scenario

Returns:

the World object

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

routing

Class SprayAndWaitRouter

```
java.lang.Object
  ↘ routing.MessageRouter
    ↘ routing.ActiveRouter
      ↘ routing.SprayAndWaitRouter
```

```
public class SprayAndWaitRouter
extends ActiveRouter
```

Implementation of Spray and wait router as depicted in *Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks* by Thrasyvoulos Spyropoulos et al.

Field Summary

static java.lang.String	BINARY_MODE identifier for the binary-mode setting ("binaryMode")
protected int	initialNrofCopies
protected boolean	isBinary
static java.lang.String	MSG_COUNT_PROPERTY Message property key
static java.lang.String	NROF_COPIES identifier for the initial number of copies setting ("nrofCopies")
static java.lang.String	SPRAYANDWAIT_NS SprayAndWait router's settings name space ("SprayAndWaitRouter")

Fields inherited from class routing.ActiveRouter

[DELETE_DELIVERED_S](#), [deleteDelivered](#), [RESPONSE_PREFIX](#), [sendingConnections](#), [TTL_CHECK_INTERVAL](#)

Fields inherited from class routing.MessageRouter

[B_SIZE_S](#), [DENIED_NO_SPACE](#), [DENIED_OLD](#), [DENIED_TTL](#), [DENIED_UNSPECIFIED](#), [MSG_TTL_S](#), [msgTtl](#), [O_MODE_FIFO](#), [O_MODE_RANDOM](#), [RCV_OK](#), [SEND_QUEUE_MODE_S](#), [TRY_LATER_BUSY](#)

Constructor Summary

	SprayAndWaitRouter(Settings s)
protected	SprayAndWaitRouter(SprayAndWaitRouter r) Copy constructor.

Method Summary

boolean

	createNewMessage(Message msg) Creates a new message to the router.
<code>protected java.util.List<Message></code>	getMessagesWithCopiesLeft() Creates and returns a list of messages this router is currently carrying and still has copies left to distribute (nrof copies > 1).
<code>Message</code>	messageTransferred(java.lang.String id, DTNHost from) This method should be called (on the receiving host) after a message was successfully transferred.
<code>int</code>	receiveMessage(Message m, DTNHost from) Try to start receiving a message from another host.
<code>SprayAndWaitRouter</code>	replicate() Creates a replicate of this router.
<code>protected void</code>	transferDone(Connection con) Called just before a transfer is finalized (by ActiveRouter.update()).
<code>void</code>	update() Checks out all sending connections to finalize the ready ones and abort those whose connection went down.

Methods inherited from class routing.[ActiveRouter](#)

[addToSendConnections](#), [canStartTransfer](#), [changedConnection](#), [checkReceiving](#), [dropExpiredMessages](#), [exchangeDeliverableMessages](#), [getConnections](#), [getMessagesForConnected](#), [getOldestMessage](#), [init](#), [isSending](#), [isTransferring](#), [makeRoomForMessage](#), [makeRoomForNewMessage](#), [requestDeliverableMessages](#), [shuffleMessages](#), [startTransfer](#), [transferAborted](#), [tryAllMessages](#), [tryAllMessagesToAllConnections](#), [tryMessagesForConnected](#), [tryMessagesToConnections](#)

Methods inherited from class routing.[MessageRouter](#)

[addApplication](#), [addMessages](#), [compareByQueueMode](#), [deleteMessage](#), [getApplications](#), [getBufferSize](#), [getFreeBufferSize](#), [getHost](#), [getMessage](#), [getMessageCollection](#), [getNrofMessages](#), [getRoutingInfo](#), [hasMessage](#), [isDeliveredMessage](#), [isIncomingMessage](#), [messageAborted](#), [putToIncomingBuffer](#), [removeFromIncomingBuffer](#), [removeFromMessages](#), [sendMessage](#), [sortByQueueMode](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

NROF_COPIES

`public static final java.lang.String NROF_COPIES`

identifier for the initial number of copies setting ("nrofCopies")

See Also:

[Constant Field Values](#)

BINARY_MODE

`public static final java.lang.String BINARY_MODE`

identifier for the binary-mode setting ("binaryMode")

See Also:

[Constant Field Values](#)**SPRAYANDWAIT_NS**

```
public static final java.lang.String SPRAYANDWAIT_NS
```

SprayAndWait router's settings name space ("SprayAndWaitRouter")

See Also:[Constant Field Values](#)**MSG_COUNT_PROPERTY**

```
public static final java.lang.String MSG_COUNT_PROPERTY
```

Message property key

See Also:[Constant Field Values](#)**initialNrofCopies**

```
protected int initialNrofCopies
```

isBinary

```
protected boolean isBinary
```

Constructor Detail**SprayAndWaitRouter**

```
public SprayAndWaitRouter(Settings s)
```

SprayAndWaitRouter

```
protected SprayAndWaitRouter(SprayAndWaitRouter r)
```

Copy constructor.

Parameters:

r - The router prototype where setting values are copied from

Method Detail**receiveMessage**

```
public int receiveMessage(Message m,  
                         DTNHost from)
```

Description copied from class: [MessageRouter](#)

Try to start receiving a message from another host.

Overrides:

[receiveMessage](#) in class [ActiveRouter](#)

Parameters:

`m` - Message to put in the receiving buffer
`from` - Who the message is from

Returns:

Value zero if the node accepted the message (RCV_OK), value less than zero if node rejected the message (e.g. DENIED_OLD), value bigger than zero if the other node should try later (e.g. TRY_LATER_BUSY).

messageTransferred

```
public Message messageTransferred(java.lang.String id,
                                  DTNHost from)
```

Description copied from class: [MessageRouter](#)

This method should be called (on the receiving host) after a message was successfully transferred. The transferred message is put to the message buffer unless this host is the final recipient of the message.

Overrides:

[messageTransferred](#) in class [ActiveRouter](#)

Parameters:

`id` - Id of the transferred message
`from` - Host the message was from (previous hop)

Returns:

The message that this host received

createNewMessage

```
public boolean createNewMessage(Message msg)
```

Description copied from class: [MessageRouter](#)

Creates a new message to the router.

Overrides:

[createNewMessage](#) in class [ActiveRouter](#)

Parameters:

`msg` - The message to create

Returns:

True if the creation succeeded, false if not (e.g. the message was too big for the buffer)

update

```
public void update()
```

Description copied from class: [ActiveRouter](#)

Checks out all sending connections to finalize the ready ones and abort those whose connection went down. Also drops messages whose TTL <= 0 (checking every one simulated minute).

Overrides:

[update](#) in class [ActiveRouter](#)

See Also:

[ActiveRouter.addToSendingConnections\(Connection\)](#)

getMessagesWithCopiesLeft

```
protected java.util.List<Message> getMessagesWithCopiesLeft()
```

Creates and returns a list of messages this router is currently carrying and still has copies left to distribute (nrof copies > 1).

Returns:

A list of messages that have copies left

transferDone

```
protected void transferDone(Connection con)
```

Called just before a transfer is finalized (by [ActiveRouter.update\(\)](#)). Reduces the number of copies we have left for a message. In binary Spray and Wait, sending host is left with floor(n/2) copies, but in standard mode, nrof copies left is reduced by one.

Overrides:

[transferDone](#) in class [ActiveRouter](#)

Parameters:

con - The connection whose transfer was finalized

replicate

```
public SprayAndWaitRouter replicate()
```

Description copied from class: [MessageRouter](#)

Creates a replicate of this router. The replicate has the same settings as this router but empty buffers and routing tables.

Specified by:

[replicate](#) in class [MessageRouter](#)

Returns:

The replicate

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

input

Class StandardEventsReader

```
java.lang.Object
└ input.StandardEventsReader
```

All Implemented Interfaces:

[ExternalEventsReader](#)

```
public class StandardEventsReader
extends java.lang.Object
implements ExternalEventsReader
```

External events reader for standard-format events (created e.g by the dtnsim2parser).

Syntax:

```
<time> <actionId> <msgId> <hostId> [<host2Id> [<size>] [<respSize>]]
```

All actions (except CONNECTION) must have first four fields. SEND, DELIVERED and ABORT actions need host2Id field too (the host who the message is/was being transferred to). CREATE action needs the additional size (of the message) field and can have also size-of-the-response field if a response to this message is requested.

CONNECTION action is followed by the two hosts which connect (or disconnect) to each other and then either "up" or "down" depending on whether the connection was created or destroyed.

Message DROP and REMOVE events can use "*" as the message ID for referring to all messages the node has in message buffer (i.e., to delete all messages).

Field Summary

static java.lang.String	ABORT Identifier of message transfer aborted event ("A")
static java.lang.String	ALL_MESSAGES_ID Message identifier to use to refer to all messages ("*")
static java.lang.String	CONNECTION Identifier of connection event ("CONN")
static java.lang.String	CONNECTION_DOWN Value identifier of connection down event ("down")
static java.lang.String	CONNECTION_UP Value identifier of connection up event ("up")
static java.lang.String	CREATE Identifier of message creation event ("C")
static java.lang.String	DELIVERED Identifier of message delivered event ("DE")
static java.lang.String	DROP Identifier of message dropped event ("DR")
static java.lang.String	REMOVE

	Identifier of message removed event ("R")
static java.lang.String SEND	Identifier of message transfer start event ("S")

Constructor Summary

[StandardEventsReader](#)(java.io.File eventsFile)

Method Summary

void close()	Closes the input file streams of the reader.
java.util.List< ExternalEvent > readEvents (int nrof)	Read events from the reader

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

CREATE

public static final java.lang.String **CREATE**

Identifier of message creation event ("C")

See Also:

[Constant Field Values](#)

SEND

public static final java.lang.String **SEND**

Identifier of message transfer start event ("S")

See Also:

[Constant Field Values](#)

DELIVERED

public static final java.lang.String **DELIVERED**

Identifier of message delivered event ("DE")

See Also:

[Constant Field Values](#)

ABORT

```
public static final java.lang.String ABORT
```

Identifier of message transfer aborted event ("A")

See Also:

[Constant Field Values](#)

DROP

```
public static final java.lang.String DROP
```

Identifier of message dropped event ("DR")

See Also:

[Constant Field Values](#)

REMOVE

```
public static final java.lang.String REMOVE
```

Identifier of message removed event ("R")

See Also:

[Constant Field Values](#)

CONNECTION

```
public static final java.lang.String CONNECTION
```

Identifier of connection event ("CONN")

See Also:

[Constant Field Values](#)

CONNECTION_DOWN

```
public static final java.lang.String CONNECTION_DOWN
```

Value identifier of connection down event ("down")

See Also:

[Constant Field Values](#)

CONNECTION_UP

```
public static final java.lang.String CONNECTION_UP
```

Value identifier of connection up event ("up")

See Also:

[Constant Field Values](#)

ALL_MESSAGES_ID

```
public static final java.lang.String ALL_MESSAGES_ID
```

Message identifier to use to refer to all messages ("*")

See Also:

[Constant Field Values](#)

Constructor Detail

StandardEventsReader

```
public StandardEventsReader(java.io.File eventsFile)
```

Method Detail

readEvents

```
public java.util.List<ExternalEvent> readEvents(int nrof)
```

Description copied from interface: [ExternalEventsReader](#)

Read events from the reader

Specified by:

[readEvents](#) in interface [ExternalEventsReader](#)

Parameters:

nrof - Maximum number of events to read

Returns:

Events in a List

close

```
public void close()
```

Description copied from interface: [ExternalEventsReader](#)

Closes the input file streams of the reader.

Specified by:

[close](#) in interface [ExternalEventsReader](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement

Class StationaryMovement

```
java.lang.Object
└ movement.MovementModel
    └ movement.StationaryMovement
```

```
public class StationaryMovement
extends MovementModel
```

A dummy stationary "movement" model where nodes do not move. Might be useful for simulations with only external connection events.

Field Summary

static java.lang.String	LOCATION_S Per node group setting for setting the location ("nodeLocation")
-------------------------	--

Fields inherited from class movement.MovementModel

```
comBus, DEF_SPEEDS, DEF_WAIT_TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime,
MOVEMENT_MODEL_NS, rng, RNG_SEED, SPEED, WAIT_TIME, WORLD_SIZE
```

Constructor Summary

[StationaryMovement\(Settings s\)](#)
Creates a new movement model based on a Settings object's settings.

[StationaryMovement\(StationaryMovement sm\)](#)
Copy constructor.

Method Summary

Coord	getInitialLocation() Returns the only location of this movement model
Path	getPath() Returns a single coordinate path (using the only possible coordinate)
double	nextPathAvailable() Returns a sim time when the next path is available.
StationaryMovement	replicate() Creates a replicate of the movement model.

Methods inherited from class movement.MovementModel

```
generateSpeed, generateWaitTime, getComBus, getMaxX, getMaxY, isActive, reset, setComBus,
toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Field Detail

LOCATION_S

```
public static final java.lang.String LOCATION_S
```

Per node group setting for setting the location ("nodeLocation")

See Also:

[Constant Field Values](#)

Constructor Detail

StationaryMovement

```
public StationaryMovement(Settings s)
```

Creates a new movement model based on a Settings object's settings.

Parameters:

s - The Settings object where the settings are read from

StationaryMovement

```
public StationaryMovement(StationaryMovement sm)
```

Copy constructor.

Parameters:

sm - The StationaryMovement prototype

Method Detail

getInitialLocation

```
public Coord getInitialLocation()
```

Returns the only location of this movement model

Specified by:

[getInitialLocation](#) in class [MovementModel](#)

Returns:

the only location of this movement model

getPath

```
public Path getPath()
```

Returns a single coordinate path (using the only possible coordinate)

Specified by:[getPath](#) in class [MovementModel](#)**Returns:**

a single coordinate path

nextPathAvailable

```
public double nextPathAvailable()
```

Description copied from class: [MovementModel](#)

Returns a sim time when the next path is available. This implementation returns a random time in future that is [MovementModel.WAIT_TIME](#) from now.

Overrides:[nextPathAvailable](#) in class [MovementModel](#)**Returns:**

The sim time when node should ask the next time for a path

replicate

```
public StationaryMovement replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Specified by:[replicate](#) in class [MovementModel](#)**Returns:**

A new movement model with the same settings as this model

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)
[PREV CLASS](#) [NEXT CLASS](#)
SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)
[FRAMES](#) [NO FRAMES](#) [All Classes](#)
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement

Interface SwitchableMovement

All Known Subinterfaces:

[TransportMovement](#)

All Known Implementing Classes:

[BusMovement](#), [BusTravellerMovement](#), [CarMovement](#), [EveningActivityMovement](#),
[HomeActivityMovement](#), [MapBasedMovement](#), [MapRouteMovement](#), [OfficeActivityMovement](#),
[RandomWalk](#), [ShortestPathMapBasedMovement](#)

```
public interface SwitchableMovement
```

Movement models to be used by ExtendedMovementModels should implement this interface

Method Summary

<code>Coord</code>	<code>getLastLocation()</code> Get the last location the getPath() of this movement model has returned
<code>boolean</code>	<code>isReady()</code> Checks if the movement model is finished doing its task and it's time to switch to the next movement model.
<code>void</code>	<code>setLocation(Coord lastWaypoint)</code> Tell the movement model what its current location is

Method Detail

setLocation

```
void setLocation(Coord lastWaypoint)
```

Tell the movement model what its current location is

Parameters:

`lastWaypoint` -

getLastLocation

```
Coord getLastLocation()
```

Get the last location the getPath() of this movement model has returned

Returns:

the last location

isReady

```
boolean isReady()
```

Checks if the movement model is finished doing its task and it's time to switch to the next movement model.
The method should be called between getPath() calls.

Returns:

true if ready

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | CONSTR | [METHOD](#)

report

Class TotalContactTimeReport

```
java.lang.Object
└ report.Report
  └ report.ContactTimesReport
    └ report.TotalContactTimeReport
```

All Implemented Interfaces:

[ConnectionListener](#), [UpdateListener](#)

```
public class TotalContactTimeReport
extends ContactTimesReport
implements UpdateListener
```

Report for total amount of contact times among hosts. Reports how long all nodes have been in contact with some other node. Supports [ContactTimesReport.GRANULARITY](#) setting. If update interval is smaller than 1.0 seconds, time stamps may start to drift. Reported values still correspond to reported times. Connections that started during the warmup period are ignored.

Nested Class Summary

Nested classes/interfaces inherited from class report.ContactTimesReport

[ContactTimesReport.ConnectionInfo](#)

Field Summary

static java.lang.String

[HEADER](#)

The header of every report file

Fields inherited from class report.ContactTimesReport

[connections](#), [granularity](#), [GRANULARITY](#)

Fields inherited from class report.Report

[DEF_PRECISION](#), [INTERVAL_SETTING](#), [INTERVALLED_FORMAT](#), [NAN](#), [out](#), [OUT_SUFFIX](#), [OUTPUT_SETTING](#), [PRECISION_SETTING](#), [REPORT_NS](#), [REPORTDIR_SETTING](#), [WARMUP_S](#), [warmupIDs](#), [warmupTime](#)

Constructor Summary

[TotalContactTimeReport\(\)](#)

Method Summary

void

[hostsDisconnected\(DTNHost host1, DTNHost host2\)](#)

Method is called when connection between hosts is disconnected.

void	<u>init()</u>	Initializes the report output.
void	<u>updated(java.util.List<DTNHost> hosts)</u>	Reports total contact time if more time than defined with setting <u>ContactTimesReport.GRANULARITY</u> has passed.

Methods inherited from class report.ContactTimesReport

[addConnection](#), [done](#), [hostsConnected](#), [increaseTimeCount](#), [removeConnection](#)

Methods inherited from class report.Report

[addWarmupID](#), [format](#), [getAverage](#), [getIntAverage](#), [getIntMedian](#), [getMedian](#), [getScenarioName](#), [getSettings](#), [getSimTime](#), [getVariance](#), [isWarmup](#), [isWarmupID](#), [newEvent](#), [removeWarmupID](#), [setPrefix](#), [write](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Field Detail

HEADER

public static final java.lang.String HEADER

The header of every report file

See Also:

[Constant Field Values](#)

Constructor Detail

TotalContactTimeReport

public [TotalContactTimeReport\(\)](#)

Method Detail

init

public void [init\(\)](#)

Description copied from class: [Report](#)

Initializes the report output. Method is called in the beginning of every new report file. Subclasses must call this method first in their own implementations of init().

Overrides:

[init](#) in class [ContactTimesReport](#)

hostsDisconnected

public void [hostsDisconnected\(DTNHost host1, DTNHost host2\)](#)

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Overrides:

[hostsDisconnected](#) in class [ContactTimesReport](#)

Parameters:

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

updated

```
public void updated(java.util.List<DTNHost> hosts)
```

Reports total contact time if more time than defined with setting [ContactTimesReport.GRANULARITY](#) has passed. Method is called on every update cycle.

Specified by:

[updated](#) in interface [UpdateListener](#)

Parameters:

hosts - A list of all hosts in the world

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class TotalEncountersReport

```
java.lang.Object
└ report.Report
  └ report.TotalEncountersReport
```

All Implemented Interfaces:[ConnectionListener](#), [UpdateListener](#)

```
public class TotalEncountersReport
extends Report
implements ConnectionListener, UpdateListener
```

A report of the distribution of how many encounters (contacts) a node has had

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[TotalEncountersReport\(\)](#)

Method Summary

void	done()	Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
int[]	getEncounters()	
void	hostsConnected(DTNHost host1, DTNHost host2)	Method is called when two hosts are connected.
void	hostsDisconnected(DTNHost host1, DTNHost host2)	Method is called when connection between hosts is disconnected.
void	setEncounters(int[] encounters)	
void	updated(java.util.List<DTNHost> hosts)	Method is called on every update cycle.

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,
getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent, removeWarmupID,
```

`setPrefix, write`**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

TotalEncountersReport

`public TotalEncountersReport()`

Method Detail

hostsConnected

`public void hostsConnected(DTNHost host1,
 DTNHost host2)`**Description copied from interface:** [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:[hostsConnected](#) in interface [ConnectionListener](#)**Parameters:**

host1 - Host that initiated the connection

host2 - Host that was connected to

hostsDisconnected

`public void hostsDisconnected(DTNHost host1,
 DTNHost host2)`**Description copied from interface:** [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:[hostsDisconnected](#) in interface [ConnectionListener](#)**Parameters:**

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

updated

`public void updated(java.util.List<DTNHost> hosts)`**Description copied from interface:** [UpdateListener](#)

Method is called on every update cycle.

Specified by:[updated](#) in interface [UpdateListener](#)

Parameters:

hosts - A list of all hosts in the world

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

getEncounters

```
public int[] getEncounters()
```

setEncounters

```
public void setEncounters(int[] encounters)
```

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

movement

Interface TransportMovement

All Superinterfaces:[SwitchableMovement](#)**All Known Implementing Classes:**[BusTravellerMovement](#), [CarMovement](#)

```
public interface TransportMovement  
extends SwitchableMovement
```

MovementModels used for transportation should implement this interface

Method Summary

void	setNextRoute (Coord nodeLocation, Coord nodeDestination)
------	---

Methods inherited from interface movement.SwitchableMovement

getLastLocation , isReady , setLocation
--

Method Detail

setNextRoute

```
void setNextRoute(Coord nodeLocation,  
Coord nodeDestination)
```

core

Class Tuple<K,V>

```
java.lang.Object
└ core.Tuple<K,V>
```

```
public class Tuple<K,V>
extends java.lang.Object
```

A generic key-value tuple.

Constructor Summary

[Tuple\(K key, V value\)](#)

Creates a new tuple.

Method Summary

K	getKey()
	Returns the key

V	getValue()
	Returns the value

java.lang.String	toString()
	Returns a string representation of the tuple

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Detail

Tuple

```
public Tuple(K key,
            V value)
```

Creates a new tuple.

Parameters:

`key` - The key of the tuple

`value` - The value of the tuple

Method Detail

getKey

```
public K getKey()
```

Returns the key

Returns:

the key

getValue

```
public V getValue()
```

Returns the value

Returns:

the value

toString

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

Returns a string representation of the tuple

Overrides:

toString in class java.lang.Object

Returns:

a string representation of the tuple

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class UniqueEncountersReport

```
java.lang.Object
└ report.Report
  └ report.UniqueEncountersReport
```

All Implemented Interfaces:

[ConnectionListener](#), [UpdateListener](#)

```
public class UniqueEncountersReport
extends Report
implements ConnectionListener, UpdateListener
```

UniqueEncountersReport class creates a report of the distribution of how many promilles of the other nodes a node has encountered.

Field Summary

Fields inherited from class report.Report

```
DEF_PRECISION, INTERVAL_SETTING, INTERVALLED_FORMAT, NAN, out, OUT_SUFFIX, OUTPUT_SETTING,
PRECISION_SETTING, REPORT_NS, REPORTDIR_SETTING, WARMUP_S, warmupIDs, warmupTime
```

Constructor Summary

[UniqueEncountersReport\(\)](#)

Method Summary

void	done() Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.
int[][]	getNodeRelationships()
void	hostsConnected(DTNHost host1, DTNHost host2) Method is called when two hosts are connected.
void	hostsDisconnected(DTNHost host1, DTNHost host2) Method is called when connection between hosts is disconnected.
void	setNodeRelationships(int[][] nodeRelationships)
void	updated(java.util.List<DTNHost> hosts) Method is called on every update cycle.

Methods inherited from class report.Report

```
addWarmupID, format, getAverage, getIntAverage, getIntMedian, getMedian, getScenarioName,  

getSettings, getSimTime, getVariance, init, isWarmup, isWarmupID, newEvent, removeWarmupID,  

setPrefix, write
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

UniqueEncountersReport

```
public UniqueEncountersReport()
```

Method Detail

hostsConnected

```
public void hostsConnected(DTNHost host1,  

                           DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when two hosts are connected.

Specified by:

[hostsConnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the connection

host2 - Host that was connected to

hostsDisconnected

```
public void hostsDisconnected(DTNHost host1,  

                             DTNHost host2)
```

Description copied from interface: [ConnectionListener](#)

Method is called when connection between hosts is disconnected.

Specified by:

[hostsDisconnected](#) in interface [ConnectionListener](#)

Parameters:

host1 - Host that initiated the disconnection

host2 - Host at the other end of the connection

updated

```
public void updated(java.util.List<DTNHost> hosts)
```

Description copied from interface: [UpdateListener](#)

Method is called on every update cycle.

Specified by:

[updated](#) in interface [UpdateListener](#)

Parameters:

hosts - A list of all hosts in the world

done

```
public void done()
```

Description copied from class: [Report](#)

Called when the simulation is done, user requested premature termination or intervalled report generating decided that it's time for the next report.

Overrides:

[done](#) in class [Report](#)

getNodeRelationships

```
public int[][][] getNodeRelationships()
```

setNodeRelationships

```
public void setNodeRelationships(int[][][] nodeRelationships)
```

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

core

Interface UpdateListener

All Known Implementing Classes:

[ContactsDuringAnICTReport](#), [EncountersVSUniqueEncountersReport](#), [EnergyLevelReport](#),
[MessageLocationReport](#), [TotalContactTimeReport](#), [TotalEncountersReport](#), [UniqueEncountersReport](#)

```
public interface UpdateListener
```

Interface for classes that want to be informed about every single update call to the World object.

NOTE: if update interval is large (if, e.g., no movement or connection simulation is needed), update listeners may not get called at all during the simulation.

Method Summary

void	updated (java.util.List< DTNHost > hosts)
	Method is called on every update cycle.

Method Detail

updated

```
void updated(java.util.List<DTNHost> hosts)
```

Method is called on every update cycle.

Parameters:

hosts - A list of all hosts in the world

core

Class VBRCConnection

```
java.lang.Object
  └ core.Connection
    └ core.VBRCConnection
```

```
public class VBRCConnection
extends Connection
```

A connection between two DTN nodes. The transmission speed is updated every round from the end point transmission speeds

Field Summary

Fields inherited from class core.Connection

bytesTransferred , fromInterface , fromNode , msgFromNode , msgOnFly , toInterface , toNode

Constructor Summary

VBRCConnection(DTNHost fromNode, NetworkInterface fromInterface, DTNHost toNode, NetworkInterface toInterface)
--

Creates a new connection between nodes and sets the connection state to "up".

Method Summary

int	getRemainingByteCount() Returns the amount of bytes to be transferred before ongoing transfer is ready or 0 if there's no ongoing transfer or it has finished already
double	getSpeed() returns the current speed of the connection
boolean	isMessageTransferred() Returns true if the current message transfer is done.
int	startTransfer(DTNHost from, Message m) Sets a message that this connection is currently transferring.
java.lang.String	toString() Returns a String presentation of the connection.
void	update() Calculate the current transmission speed from the information given by the interfaces, and calculate the missing data amount.

Methods inherited from class core.Connection

abortTransfer , clearMsgOnFly , finalizeTransfer , getMessage , getOtherInterface , getOtherNode , getTotalBytesTransferred , isInitiator , isReadyForTransfer , isUp , setUpState
--

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail**VBRConnection**

```
public VBRConnection(DTNHost fromNode,
                     NetworkInterface fromInterface,
                     DTNHost toNode,
                     NetworkInterface toInterface)
```

Creates a new connection between nodes and sets the connection state to "up".

Parameters:

- fromNode - The node that initiated the connection
- fromInterface - The interface that initiated the connection
- toNode - The node in the other side of the connection
- toInterface - The interface in the other side of the connection

Method Detail**startTransfer**

```
public int startTransfer(DTNHost from,
                        Message m)
```

Sets a message that this connection is currently transferring. If message passing is controlled by external events, this method is not needed (but then e.g. [Connection.finalizeTransfer\(\)](#) and [isMessageTransferred\(\)](#) will not work either). Only a one message at a time can be transferred using one connection.

Specified by:

[startTransfer](#) in class [Connection](#)

Parameters:

- from - The host sending the message
- m - The message

Returns:

The value returned by [MessageRouter.receiveMessage\(Message, DTNHost\)](#)

update

```
public void update()
```

Calculate the current transmission speed from the information given by the interfaces, and calculate the missing data amount.

Overrides:

[update](#) in class [Connection](#)

getSpeed

```
public double getSpeed()
```

returns the current speed of the connection

Specified by:

[getSpeed](#) in class [Connection](#)

getRemainingByteCount

public int **getRemainingByteCount()**

Returns the amount of bytes to be transferred before ongoing transfer is ready or 0 if there's no ongoing transfer or it has finished already

Specified by:

[getRemainingByteCount](#) in class [Connection](#)

Returns:

the amount of bytes to be transferred

isMessageTransferred

public boolean **isMessageTransferred()**

Returns true if the current message transfer is done.

Specified by:

[isMessageTransferred](#) in class [Connection](#)

Returns:

True if the transfer is done, false if not

toString

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

Returns a String presentation of the connection.

Overrides:

[toString](#) in class [Connection](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

input

Class WKTMapReader

```
java.lang.Object
└ input.WKTReader
    └ input.WKTMapReader
```

```
public class WKTMapReader
extends WKTReader
```

"Well-known text syntax" map data reader.

Note: Understands only `LINESTRINGS` and `MULTILINESTRINGS`. Skips all `POINT` data. Other data causes `IOException`.

Field Summary

Fields inherited from class input.WKTReader

[LINESTRING](#), [MULTILINESTRING](#), [POINT](#)

Constructor Summary

[WKTMapReader\(boolean bidi\)](#)

Constructor.

Method Summary

void addPaths(java.io.File file, int type) Add paths to the map and adds given type to all nodes' type.
void addPaths(java.io.Reader input, int nodeType) Add paths to current path set.
SimMap getMap() Returns new a SimMap that is based on the read map
java.util.Collection< MapNode > getNodes() Returns the map nodes that were read in a collection
java.util.Map< Coord , MapNode > getNodesHash() Returns the original Map object that was used to read the map
void setBidirectional(boolean bidi) Sets bidirectional paths on/off.

Methods inherited from class input.WKTReader

[init](#), [isDone](#), [isKnownType](#), [nextType](#), [parseLineString](#), [parseMultilinestring](#), [parsePoint](#), [readLines](#), [readNestedContents](#), [readNestedContents](#), [readPoints](#), [readPoints](#), [readWord](#), [setDone](#), [skipAllWhitespace](#), [skipUntil](#)

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

WKTMapReader

```
public WKTMapReader(boolean bidi)
```

Constructor. Creates a new WKT reader ready for addPaths() calls.

Parameters:

bidi - If true, all read paths are set bidirectional (i.e. if node A is a neighbor of node B, node B is also a neighbor of node A).

Method Detail

setBidirectional

```
public void setBidirectional(boolean bidi)
```

Sets bidirectional paths on/off.

Parameters:

bidi - If true, all paths are set bidirectional (false -> not)

getNodes

```
public java.util.Collection<MapNode> getNodes()
```

Returns the map nodes that were read in a collection

Returns:

the map nodes that were read in a collection

getNodesHash

```
public java.util.Map<Coord, MapNode> getNodesHash()
```

Returns the original Map object that was used to read the map

Returns:

the original Map object that was used to read the map

getMap

```
public SimMap getMap()
```

Returns new a SimMap that is based on the read map

Returns:

new a SimMap that is based on the read map

addPaths

```
public void addPaths(java.io.File file,
                     int type)
                     throws java.io.IOException
```

Adds paths to the map and adds given type to all nodes' type.

Parameters:

file - The file where the WKT data is read from
 type - The type to use (integer value, see class [MapNode](#)))

Throws:

`java.io.IOException` - If something went wrong while reading the file

addPaths

```
public void addPaths(java.io.Reader input,
                     int nodeType)
                     throws java.io.IOException
```

Add paths to current path set. Adding paths multiple times has the same result as concatenating the data before adding it.

Parameters:

input - Reader where the WKT data is read from
 nodeType - The type to use (integer value, see class [MapNode](#)))

Throws:

`java.io.IOException` - if something went wrong with reading from the input

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

input

Class WKTReader

```
java.lang.Object
└─ input.WKTReader
```

Direct Known Subclasses:[WKTMapReader](#)

```
public class WKTReader
extends java.lang.Object
```

Class for reading "Well-known text syntax" files. See e.g. [Wikipedia](#) for WKT syntax details. For example, [Open JUMP](#) GIS program can save compatible data from many other formats.

Field Summary

static java.lang.String	LINESTRING known WKT type LINESTRING
static java.lang.String	MULTILINESTRING known WKT type MULTILINESTRING
static java.lang.String	POINT known WKT type POINT

Constructor Summary

[WKTReader\(\)](#)

Method Summary

protected void	init(java.io.Reader input) Initialize the reader to use a certain input reader
protected boolean	isDone() Returns true if the whole file has been read
protected boolean	isKnownType(java.lang.String type) Returns true if type is one of the known WKT types
protected java.lang.String	nextType() Returns the next type read from the reader given at init or null if no more types can be read
protected java.util.List< Coord >	parseLineString(java.lang.String line) Parses coordinate tuples from "LINESTRING" lines
protected java.util.List<java.util.List< Coord >>	parseMultilinestring() Parses a MULTILINESTRING statement that has nested linestrings from the current reader
protected Coord	

	parsePoint() Parses a WKT point data from the initialized reader
java.util.List<java.util.List< Coord >>	readLines(java.io.File file) Read line (LINESTRING) data from a file
java.lang.String	readNestedContents() Returns nested contents from the reader given at init
java.lang.String	readNestedContents(java.io.Reader r) Reads everything from the first opening parenthesis until line that ends to a closing parenthesis and returns the contents in one string
java.util.List< Coord >	readPoints(java.io.File file) Read point data from a file
java.util.List< Coord >	readPoints(java.io.Reader r) Read point data from a Reader
protected java.lang.String	readWord(java.io.Reader r) Reads a "word", ie whitespace delimited string of characters, from the reader
protected void	setDone(boolean done) Sets the "is file read" state
protected char	skipAllWhitespace(java.io.Reader r) Skips all consecutive whitespace characters from reader
protected void	skipUntil(java.io.Reader r, char until) Reads and skips all characters until character "until" is read or end of stream is reached.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

LINESTRING

public static final java.lang.String **LINESTRING**

known WKT type LINESTRING

See Also:[Constant Field Values](#)

MULTILINESTRING

public static final java.lang.String **MULTILINESTRING**

known WKT type MULTILINESTRING

See Also:[Constant Field Values](#)

POINT

```
public static final java.lang.String POINT
```

known WKT type POINT

See Also:

[Constant Field Values](#)

Constructor Detail

WKTReader

```
public WKTReader()
```

Method Detail

readPoints

```
public java.util.List<Coord> readPoints(java.io.File file)
throws java.io.IOException
```

Read point data from a file

Parameters:

file - The file to read points from

Returns:

A list of coordinates read from the file

Throws:

java.io.IOException - if something went wrong while reading

readPoints

```
public java.util.List<Coord> readPoints(java.io.Reader r)
throws java.io.IOException
```

Read point data from a Reader

Parameters:

r - The Reader to read points from

Returns:

A list of coordinates that were read

Throws:

java.io.IOException - if something went wrong while reading

readLines

```
public java.util.List<java.util.List<Coord>> readLines(java.io.File file)
throws java.io.IOException
```

Read line (LINESTRING) data from a file

Parameters:

file - The file to read data from

Returns:

A list of coordinate lists read from the file

Throws:

`java.io.IOException` - if something went wrong while reading

init

```
protected void init(java.io.Reader input)
```

Initialize the reader to use a certain input reader

Parameters:

`input` - The input to use

nextType

```
protected java.lang.String nextType()
    throws java.io.IOException
```

Returns the next type read from the reader given at init or null if no more types can be read

Returns:

the next type read from the reader given at init

Throws:

`java.io.IOException`

isKnownType

```
protected boolean isKnownType(java.lang.String type)
```

Returns true if type is one of the known WKT types

Parameters:

`type` - The type to check

Returns:

true if type is one of the known WKT types

readWord

```
protected java.lang.String readWord(java.io.Reader r)
    throws java.io.IOException
```

Reads a "word", ie whitespace delimited string of characters, from the reader

Parameters:

`r` - Reader to read the characters from

Returns:

The word that was read (or empty string if nothing was read)

Throws:

`java.io.IOException`

parseMultilinestring

```
protected java.util.List<java.util.List<Coord>> parseMultilinestring()
    throws java.io.IOException
```

Parses a MULTILINESTRING statement that has nested linestrings from the current reader

Returns:

List of parsed Coord lists

Throws:

java.io.IOException

parsePoint

```
protected Coord parsePoint()
    throws java.io.IOException
```

Parses a WKT point data from the initialized reader

Returns:

Point data as a Coordinate

Throws:

java.io.IOException - if couldn't parse coordinate values

skipUntil

```
protected void skipUntil(java.io.Reader r,
    char until)
    throws java.io.IOException
```

Reads and skips all characters until character "until" is read or end of stream is reached. Also the expected character is discarded.

Parameters:

r - Reader to read characters from

until - What character to expect

Throws:

java.io.IOException

skipAllWhitespace

```
protected char skipAllWhitespace(java.io.Reader r)
    throws java.io.IOException
```

Skips all consecutive whitespace characters from reader

Parameters:

r - Reader where the whitespace is skipped

Returns:

First non-whitespace character read from the reader

Throws:

java.io.IOException

readNestedContents

```
public java.lang.String readNestedContents(java.io.Reader r)
    throws java.io.IOException
```

Reads everything from the first opening parenthesis until line that ends to a closing parenthesis and returns the contents in one string

Parameters:

r - Reader to read the input from

Returns:

The text between the parentheses

Throws:

java.io.IOException

readNestedContents

```
public java.lang.String readNestedContents()
                      throws java.io.IOException
```

Returns nested contents from the reader given at init

Returns:

nested contents from the reader given at init

Throws:

java.io.IOException

See Also:

[readNestedContents\(Reader\)](#)

parseLineString

```
protected java.util.List<Coord> parseLineString(java.lang.String line)
```

Parses coordinate tuples from "LINESTRING" lines

Parameters:

line - String that contains the whole "LINESTRING"'s content

Returns:

List of coordinates parsed from the linestring

isDone

```
protected boolean isDone()
```

Returns true if the whole file has been read

Returns:

true if the whole file has been read

setDone

```
protected void setDone(boolean done)
```

Sets the "is file read" state

Parameters:

done - If true, reading is done

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

movement

Class WorkingDayMovement

```
java.lang.Object
  └── movement.MovementModel
    └── movement.ExtendedMovementModel
      └── movement.WorkingDayMovement
```

```
public class WorkingDayMovement
extends ExtendedMovementModel
```

This movement model makes use of several other movement models to simulate movement with daily routines. People wake up in the morning, go to work, go shopping or similar activities in the evening and finally go home to sleep.

Field Summary

static java.lang.String	PROBABILITY TO GO SHOPPING SETTING
-------------------------	--

static java.lang.String	PROBABILITY TO OWN CAR SETTING
-------------------------	--

Fields inherited from class movement.MovementModel

comBus, DEF SPEEDS, DEF WAIT TIMES, maxSpeed, maxWaitTime, minSpeed, minWaitTime, MOVEMENT MODEL NS, rng, RNG SEED, SPEED, WAIT TIME, WORLD SIZE

Constructor Summary

WorkingDayMovement(Settings settings)

Creates a new instance of WorkingDayMovement

WorkingDayMovement(WorkingDayMovement proto)
--

Creates a new instance of WorkingDayMovement from a prototype

Method Summary

Coord	getHomeLocation()
-----------------------	-----------------------------------

Coord	getInitialLocation()
-----------------------	--------------------------------------

Returns a new initial placement for a node

Coord	getOfficeLocation()
-----------------------	-------------------------------------

Coord	getShoppingLocation()
-----------------------	---------------------------------------

boolean	newOrders()
---------	-----------------------------

Method is called between each getPath() request when the current MM is ready (isReady() method returns true).

MovementModel	replicate()
-------------------------------	-----------------------------

Creates a replicate of the movement model.

Methods inherited from class movement.[ExtendedMovementModel](#)

[generateWaitTime](#), [getCurrentMovementModel](#), [getPath](#), [setCurrentMovementModel](#)

Methods inherited from class movement.[MovementModel](#)

[generateSpeed](#), [getComBus](#), [getMaxX](#), [getMaxY](#), [isActive](#), [nextPathAvailable](#), [reset](#), [setComBus](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

PROBABILITY_TO_OWN_CAR_SETTING

public static final java.lang.String **PROBABILITY_TO_OWN_CAR_SETTING**

See Also:

[Constant Field Values](#)

PROBABILITY_TO_GO_SHOPPING_SETTING

public static final java.lang.String **PROBABILITY_TO_GO_SHOPPING_SETTING**

See Also:

[Constant Field Values](#)

Constructor Detail

WorkingDayMovement

public **WorkingDayMovement**([Settings](#) settings)

Creates a new instance of WorkingDayMovement

Parameters:

settings -

WorkingDayMovement

public **WorkingDayMovement**([WorkingDayMovement](#) proto)

Creates a new instance of WorkingDayMovement from a prototype

Parameters:

proto -

Method Detail

newOrders

```
public boolean newOrders()
```

Description copied from class: [ExtendedMovementModel](#)

Method is called between each getPath() request when the current MM is ready (isReady() method returns true). Subclasses should implement all changes of state that need to be made here, for example switching mobility model, etc.

Specified by:

[newOrders](#) in class [ExtendedMovementModel](#)

Returns:

true if success

getInitialLocation

```
public Coord getInitialLocation()
```

Description copied from class: [MovementModel](#)

Returns a new initial placement for a node

Specified by:

[getInitialLocation](#) in class [MovementModel](#)

Returns:

The initial coordinates for a node

replicate

```
public MovementModel replicate()
```

Description copied from class: [MovementModel](#)

Creates a replicate of the movement model.

Specified by:

[replicate](#) in class [MovementModel](#)

Returns:

A new movement model with the same settings as this model

getOfficeLocation

```
public Coord getOfficeLocation()
```

getHomeLocation

```
public Coord getHomeLocation()
```

getShoppingLocation

```
public Coord getShoppingLocation()
```

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

core

Class World

```
java.lang.Object
└ core.World
```

```
public class World
extends java.lang.Object
```

World contains all the nodes and is responsible for updating their location and connections.

Field Summary

static java.lang.String	CELL_SIZE_MULT_S Cell based optimization cell size multiplier -setting id ("cellSizeMult").
static int	DEF_CON_CELL_SIZE_MULT default value for cell size multiplier (5)
static boolean	DEF_RANDOMIZE_UPDATES should the update order of nodes be randomized -setting's default value (true)
static java.lang.String	RANDOMIZE_UPDATES_S Should the order of node updates be different (random) within every update step -setting id ("randomizeUpdateOrder").
static java.lang.String	SETTINGS_NS namespace of optimization settings ("Optimization")

Constructor Summary

```
World(java.util.List<DTNHost> hosts, int sizeX, int sizeY, double updateInterval,
java.util.List<UpdateListener> updateListeners, boolean simulateConnections,
java.util.List<EventQueue> eventQueues)
    Constructor.
```

Method Summary

void	cancelSim() Asynchronously cancels the currently running simulation
java.util.List<DTNHost>	getHosts() Returns the hosts in a list
DTNHost	getNodeByAddress(int address) Returns a node from the world by its address
int	getSizeX() Returns the x-size (width) of the world
int	getSizeY() Returns the y-size (height) of the world
void	scheduleUpdate(double simTime)

	Schedules an update request to all nodes to happen at the specified simulation time.
void setNextEventQueue()	Goes through all event Queues and sets the event queue that has the next event.
void update()	Update (move, connect, disconnect etc.) all hosts in the world.
void warmupMovementModel(double time)	Moves hosts in the world for the time given time initialize host positions properly.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

SETTINGS_NS

`public static final java.lang.String SETTINGS_NS`

namespace of optimization settings ("Optimization")

See Also:

[Constant Field Values](#)

CELL_SIZE_MULT_S

`public static final java.lang.String CELL_SIZE_MULT_S`

Cell based optimization cell size multiplier -setting id ("cellSizeMult"). Single ConnectivityCell's size is the biggest radio range times this. Larger values save memory and decrease startup time but may result in slower simulation. Default value is [DEF_CON_CELL_SIZE_MULT](#). Smallest accepted value is 2.

See Also:

[ConnectivityGrid](#), [Constant Field Values](#)

RANDOMIZE_UPDATES_S

`public static final java.lang.String RANDOMIZE_UPDATES_S`

Should the order of node updates be different (random) within every update step -setting id ("randomizeUpdateOrder"). Boolean (true/false) variable. Default is @link [DEF_RANDOMIZE_UPDATES](#).

See Also:

[Constant Field Values](#)

DEF_CON_CELL_SIZE_MULT

`public static final int DEF_CON_CELL_SIZE_MULT`

default value for cell size multiplier (5)

See Also:[Constant Field Values](#)**DEF_RANDOMIZE_UPDATES**

```
public static final boolean DEF_RANDOMIZE_UPDATES
```

should the update order of nodes be randomized -setting's default value (true)

See Also:[Constant Field Values](#)**Constructor Detail****World**

```
public World(java.util.List<DTNHost> hosts,
            int sizeX,
            int sizeY,
            double updateInterval,
            java.util.List<UpdateListener> updateListeners,
            boolean simulateConnections,
            java.util.List<EventQueue> eventQueues)
```

Constructor.

Method Detail**warmupMovementModel**

```
public void warmupMovementModel(double time)
```

Moves hosts in the world for the time given time initialize host positions properly. SimClock must be set to -time before calling this method.

Parameters:

`time` - The total time (seconds) to move

setNextEventQueue

```
public void setNextEventQueue()
```

Goes through all event Queues and sets the event queue that has the next event.

update

```
public void update()
```

Update (move, connect, disconnect etc.) all hosts in the world. Runs all external events that are due between the time when this method is called and after one update interval.

cancelSim

```
public void cancelSim()
```

Asynchronously cancels the currently running simulation

getHosts

```
public java.util.List<DTNHost> getHosts()
```

Returns the hosts in a list

Returns:

the hosts in a list

getSizeX

```
public int getSizeX()
```

Returns the x-size (width) of the world

Returns:

the x-size (width) of the world

getSizeY

```
public int getSizeY()
```

Returns the y-size (height) of the world

Returns:

the y-size (height) of the world

getNodeByAddress

```
public DTNHost getNodeByAddress(int address)
```

Returns a node from the world by its address

Parameters:

address - The address of the node

Returns:

The requested node or null if it wasn't found

scheduleUpdate

```
public void scheduleUpdate(double simTime)
```

Schedules an update request to all nodes to happen at the specified simulation time.

Parameters:

simTime - The time of the update

Hierarchy For Package movement

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - movement.[ActivenessHandler](#)
 - movement.[BusControlSystem](#)
 - movement.[EveningActivityControlSystem](#)
 - movement.[EveningTrip](#)
 - movement.[MovementModel](#)
 - movement.[ExtendedMovementModel](#)
 - movement.[WorkingDayMovement](#)
 - movement.[ExternalMovement](#)
 - movement.[LinearFormation](#)
 - movement.[MapBasedMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[BusTravellerMovement](#) (implements movement.[SwitchableMovement](#), movement.[TransportMovement](#))
 - movement.[CarMovement](#) (implements movement.[SwitchableMovement](#), movement.[TransportMovement](#))
 - movement.[EveningActivityMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[HomeActivityMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[MapRouteMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[BusMovement](#)
 - movement.[OfficeActivityMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[ShortestPathMapBasedMovement](#) (implements movement.[SwitchableMovement](#))
 - movement.[RandomWalk](#) (implements movement.[SwitchableMovement](#))
 - movement.[RandomWaypoint](#)
 - movement.[ClusterMovement](#)
 - movement.[StationaryMovement](#)
 - movement.[Path](#)

Interface Hierarchy

- movement.[SwitchableMovement](#)
 - movement.[TransportMovement](#)

All Classes

[ActivenessHandler](#)
[ActiveRouter](#)
[AdjacencyGraphvizReport](#)
[Application](#)
[ApplicationListener](#)
[BinaryEventsReader](#)
[BusControlSystem](#)
[BusMovement](#)
[BusTravellerMovement](#)
[CarMovement](#)
[CBRConnection](#)
[ClusterMovement](#)
[Connection](#)
[ConnectionEvent](#)
[ConnectionListener](#)
[ConnectivityDtnsim2Report](#)
[ConnectivityGrid](#)
[ConnectivityONEReport](#)
[ConnectivityOptimizer](#)
[ContactsDuringAnICTReport](#)
[ContactsPerHourReport](#)
[ContactTimesReport](#)
[Coord](#)
[CreatedMessagesReport](#)
[Debug](#)
[DeliveredMessagesReport](#)
[DijkstraPathFinder](#)
[DirectDeliveryRouter](#)
[DistanceDelayReport](#)
[DTN2Events](#)
[DTN2Manager](#)
[DTN2Manager.EIDHost](#)
[DTN2Reporter](#)
[DTNHost](#)
[DTNSim](#)
[DTNSimGUI](#)
[DTNSimTextUI](#)
[DTNSimUI](#)
[EncountersVSUniqueEncountersReport](#)
[EnergyAwareRouter](#)
[EnergyLevelReport](#)
[EpidemicOracleRouter](#)
[EpidemicRouter](#)
[EveningActivityControlSystem](#)
[EveningActivityMovement](#)
[EveningTrip](#)
[EventLogControl](#)
[EventLogControlPanel](#)
[EventLogPanel](#)
[EventLogReport](#)
[EventQueue](#)
[EventQueueHandler](#)
[ExtendedMovementModel](#)
[ExternalEvent](#)
[ExternalEventsQueue](#)
[ExternalEventsReader](#)
[ExternalMovement](#)
[ExternalMovementReader](#)
[FirstContactRouter](#)

[GUIControls](#)
[HomeActivityMovement](#)
[InfoPanel](#)
[InterContactTimesReport](#)
[InterferenceLimitedInterface](#)
[LinearFormation](#)
[MainWindow](#)
[MapBasedMovement](#)
[MapGraphic](#)
[MapNode](#)
[MapRoute](#)
[MapRouteMovement](#)
[MaxPropDijkstra](#)
[MaxPropRouter](#)
[MaxPropRouterWithEstimation](#)
[MeetingProbabilitySet](#)
[Message](#)
[MessageBurstGenerator](#)
[MessageCreateEvent](#)
[MessageDelayReport](#)
[MessageDeleteEvent](#)
[MessageDeliveryReport](#)
[MessageEvent](#)
[MessageEventGenerator](#)
[MessageGraphic](#)
[MessageGraphvizReport](#)
[MessageListener](#)
[MessageLocationReport](#)
[MessageRelayEvent](#)
[MessageReport](#)
[MessageRouter](#)
[MessageStatsReport](#)
[ModuleCommunicationBus](#)
[ModuleCommunicationListener](#)
[MovementListener](#)
[MovementModel](#)
[MovementNs2Report](#)
[NetworkInterface](#)
[NodeChooser](#)
[NodeGraphic](#)
[OfficeActivityMovement](#)
[OneFromEachMessageGenerator](#)
[OneToEachMessageGenerator](#)
[ParetoRNG](#)
[PassiveRouter](#)
[Path](#)
[PathGraphic](#)
[PingApplication](#)
[PingAppReporter](#)
[PlayField](#)
[PlayFieldGraphic](#)
[PointsOfInterest](#)
[ProphetRouter](#)
[ProphetRouterWithEstimation](#)
[RandomWalk](#)
[RandomWaypoint](#)
[Report](#)
[RoutingInfo](#)
[RoutingInfoWindow](#)
[ScaleReferenceGraphic](#)
[ScheduleDijkstra](#)

[ScheduledUpdatesQueue](#)[ScheduleEntry](#)[ScheduleOracle](#)[Settings](#)[SettingsError](#)[ShortestPathMapBasedMovement](#)[SimClock](#)[SimError](#)[SimMap](#)[SimMenuBar](#)[SimpleBroadcastInterface](#)[SimScenario](#)[SprayAndWaitRouter](#)[StandardEventsReader](#)[StationaryMovement](#)[SwitchableMovement](#)[TotalContactTimeReport](#)[TotalEncountersReport](#)[TransportMovement](#)[Tuple](#)[UniqueEncountersReport](#)[UpdateListener](#)[VBRConnection](#)[WKTMapReader](#)[WKTRader](#)[WorkingDayMovement](#)[World](#)

Constant Field Values

Contents

- [applications.*](#)
- [core.*](#)
- [gui.*](#)
- [input.*](#)
- [movement.*](#)
- [report.*](#)
- [routing.*](#)
- [ui.*](#)

applications.*

[applications.PingApplication](#)

public static final java.lang.String	APP_ID	"fi.tkk.netlab.PingApplication"
public static final java.lang.String	PING_DEST_RANGE	"destinationRange"
public static final java.lang.String	PING_INTERVAL	"interval"
public static final java.lang.String	PING_OFFSET	"offset"
public static final java.lang.String	PING_PASSIVE	"passive"
public static final java.lang.String	PING_PING_SIZE	"pingSize"
public static final java.lang.String	PING_PONG_SIZE	"pongSize"
public static final java.lang.String	PING_SEED	"seed"

core.*

[core.DTNSim](#)

public static final java.lang.String	BATCH_MODE_FLAG	" -b"
public static final java.lang.String	RANGE_DELIMETER	" :"
public static final java.lang.String	RESET_METHOD_NAME	"reset"

[core.Message](#)

public static final int	INFINITE_TTL	-1
-------------------------	------------------------------	----

[core.NetworkInterface](#)

public static final java.lang.String	RANGE_ID	"Network.radioRange"
public static final java.lang.String	SCAN_INTERVAL_ID	"Network.scanInterval"
public static final java.lang.String	SCAN_INTERVAL_S	"scanInterval"
public static final java.lang.String	SPEED_ID	"Network.speed"
public static final java.lang.String	TRANSMIT_RANGE_S	"transmitRange"
public static final java.lang.String	TRANSMIT_SPEED_S	"transmitSpeed"

[core.Settings](#)

public static final java.lang.String	DEF_SETTINGS_FILE	"default_settings.txt"
public static final java.lang.String	FILL_DELIMITER	"%%"
public static final java.lang.String	SETTING_OUTPUT_S	"Settings.output"

[core.SimScenario](#)

public static final java.lang.String	APPCOUNT_S	"nrofApplications"
--------------------------------------	----------------------------	--------------------

public static final java.lang.String	APPTYPE_NS	"Application"
public static final java.lang.String	APPTYPE_S	"type"
public static final java.lang.String	END_TIME_S	"endTime"
public static final java.lang.String	GAPPNAME_S	"application"
public static final java.lang.String	GROUP_ID_S	"groupID"
public static final java.lang.String	GROUP_NS	"Group"
public static final java.lang.String	INTERFACENAME_S	"interface"
public static final java.lang.String	INTNAME_S	"name"
public static final java.lang.String	INTTYPE_NS	"Interface"
public static final java.lang.String	INTTYPE_S	"type"
public static final java.lang.String	MOVEMENT_MODEL_S	"movementModel"
public static final java.lang.String	NAME_S	"name"
public static final java.lang.String	NROF_GROUPS_S	"nrofHostGroups"
public static final java.lang.String	NROF_HOSTS_S	"nrofHosts"
public static final java.lang.String	NROF_INTERF_S	"nrofInterfaces"
public static final java.lang.String	NROF_INTTYPES_S	"nrofInterfaceTypes"
public static final java.lang.String	ROUTER_S	"router"
public static final java.lang.String	SCAN_INTERVAL_S	"scanInterval"
public static final java.lang.String	SCENARIO_NS	"Scenario"
public static final java.lang.String	SIM_CON_S	"simulateConnections"
public static final java.lang.String	UP_INT_S	"updateInterval"

core.World

public static final java.lang.String	CELL_SIZE_MULT_S	"cellSizeMult"
public static final int	DEF_CON_CELL_SIZE_MULT	5
public static final boolean	DEF_RANDOMIZE_UPDATES	true
public static final java.lang.String	RANDOMIZE_UPDATES_S	"randomizeUpdateOrder"
public static final java.lang.String	SETTINGS_NS	"Optimization"

gui.***gui.EventLogPanel**

public static final int	LOG_UP_INTERVAL	500
-------------------------	---------------------------------	-----

gui.GUIControls

public static final int	FW_SPEED_INDEX	7
public static final int	INITIAL_SPEED_SELECTION	3
public static final double	ZOOM_MAX	10.0
public static final double	ZOOM_MIN	0.0010

gui.NodeChooser

public static final int	MAX_NODE_COUNT	500
-------------------------	--------------------------------	-----

gui.SimMenuBar

public static final java.lang.String	ABOUT_TEXT	"Copyright (C) 2007 TKK/Netlab\n\nThis program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.\n\nThis program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.\n\nYou should have received a copy of the GNU General Public License along with this program. If not, see < http://www.gnu.org/licenses/ >.\n\nMap data copyright: Maanmittauslaitos, 2007"
public static final java.lang.String	ABOUT_TITLE	"about ONE"

input.***input.BinaryEventsReader**

public static final java.lang.String	BINARY_EXT	".binee"
--------------------------------------	----------------------------	----------

input.EventQueueHandler

public static final java.lang.String	CLASS_PACKAGE	"input"
public static final java.lang.String	CLASS_SETTING	"class"
public static final java.lang.String	NROF_SETTING	"nrof"
public static final java.lang.String	PATH_SETTING	"filePath"
public static final java.lang.String	PRELOAD_SETTING	"nrofPreload"
public static final java.lang.String	SETTINGS_NAMESPACE	"Events"

input.ExternalEventsQueue

public static final int	DEFAULT_NROF_PRELOAD	500
public static final java.lang.String	PATH_SETTING	"filePath"
public static final java.lang.String	PRELOAD_SETTING	"nrofPreload"
public static final java.lang.String	SETTINGS_NAMESPACE	"ExternalEvents"

input.ExternalMovementReader

public static final java.lang.String	COMMENT_PREFIX	"#"
--------------------------------------	--------------------------------	-----

input.MessageEventGenerator

public static final java.lang.String	HOST_RANGE_S	"hosts"
public static final java.lang.String	MESSAGE_ID_PREFIX_S	"prefix"
public static final java.lang.String	MESSAGE_INTERVAL_S	"interval"
public static final java.lang.String	MESSAGE_SIZE_S	"size"
public static final java.lang.String	MESSAGE_TIME_S	"time"
public static final java.lang.String	TO_HOST_RANGE_S	"tohosts"

input.MessageRelayEvent

public static final int	ABORTED	3
public static final int	SENDING	1
public static final int	TRANSFERRED	2

input.StandardEventsReader

public static final java.lang.String	ABORT	"A"
public static final java.lang.String	ALL_MESSAGES_ID	"*"
public static final java.lang.String	CONNECTION	"CONN"
public static final java.lang.String	CONNECTION_DOWN	"down"
public static final java.lang.String	CONNECTION_UP	"up"
public static final java.lang.String	CREATE	"C"
public static final java.lang.String	DELIVERED	"DE"
public static final java.lang.String	DROP	"DR"
public static final java.lang.String	REMOVE	"R"
public static final java.lang.String	SEND	"S"

input.WKTReader

public static final java.lang.String	LINESTRING	"LINESTRING"
public static final java.lang.String	MULTILINESTRING	"MULTILINESTRING"
public static final java.lang.String	POINT	"POINT"

movement.*

[movement.ActiveHandler](#)

public static final java.lang.String	ACTIVE_TIMES_S	"activeTimes"
--------------------------------------	--------------------------------	---------------

[movement.BusControlSystem](#)

public static final java.lang.String	BUS_CONTROL_SYSTEM_NR	"busControlSystemNr"
--------------------------------------	---------------------------------------	----------------------

[movement.BusTravellerMovement](#)

public static final java.lang.String	PROBABILITIES_STRING	"probs"
public static final java.lang.String	PROBABILITY_TAKE_OTHER_BUS	"probTakeOtherBus"
public static final int	STATE_DECIDED_TO_ENTER_A_BUS	1
public static final int	STATE_TRAVELLING_ON_BUS	2
public static final int	STATE_WAITING_FOR_BUS	0
public static final int	STATE_WALKING_ELSEWHERE	3

[movement.ClusterMovement](#)

public static final java.lang.String	CLUSTER_CENTER	"clusterCenter"
public static final java.lang.String	CLUSTER_RANGE	"clusterRange"

[movement.EveningActivityMovement](#)

public static final java.lang.String	EVENING_ACTIVITY_CONTROL_SYSTEM_NR_SETTING	"shoppingControlSystemNr"
public static final java.lang.String	MAX_GROUP_SIZE_SETTING	"maxGroupSize"
public static final java.lang.String	MAX_WAIT_TIME_SETTING	"maxAfterShoppingStopTime"
public static final java.lang.String	MEETING_SPOTS_FILE_SETTING	"meetingSpotsFile"
public static final java.lang.String	MIN_GROUP_SIZE_SETTING	"minGroupSize"
public static final java.lang.String	MIN_WAIT_TIME_SETTING	"minAfterShoppingStopTime"
public static final java.lang.String	NR_OF_MEETING_SPOTS_SETTING	"nrOfMeetingSpots"

[movement.ExternalMovement](#)

public static final java.lang.String	EXTERNAL_MOVEMENT_NS	"ExternalMovement"
public static final java.lang.String	MOVEMENT_FILE_S	"file"
public static final java.lang.String	NROF_PRELOAD_S	"nrofPreload"

[movement.HomeActivityMovement](#)

public static final java.lang.String	HOME_LOCATIONS_FILE_SETTING	"homeLocationsFile"
public static final java.lang.String	STD_FOR_TIME_DIFF_SETTING	"timeDiffSTD"

[movement.LinearFormation](#)

public static final java.lang.String	END_LOCATION_S	"endLocation"
public static final java.lang.String	LINEAR_FORMATION_NS	"LinearFormation."
public static final java.lang.String	START_LOCATION_S	"startLocation"

[movement.MapBasedMovement](#)

public static final java.lang.String	FILE_S	"mapFile"
public static final java.lang.String	MAP_BASE_MOVEMENT_NS	"MapBasedMovement"
public static final java.lang.String	MAP_SELECT_S	"okMaps"
public static final java.lang.String	NROF_FILES_S	"nrofMapFiles"

[movement.MapRouteMovement](#)

public static final java.lang.String	ROUTE_FILE_S	"routeFile"
public static final java.lang.String	ROUTE_FIRST_STOP_S	"routeFirstStop"
public static final java.lang.String	ROUTE_TYPE_S	"routeType"

movement.MovementModel

public static final java.lang.String	MOVEMENT_MODEL_NS	"MovementModel"
public static final java.lang.String	RNG_SEED	"rngSeed"
public static final java.lang.String	SPEED	"speed"
public static final java.lang.String	WAIT_TIME	"waitTime"
public static final java.lang.String	WORLD_SIZE	"worldSize"

movement.OfficeActivityMovement

public static final java.lang.String	NR_OF_OFFICES_SETTING	"nrOfOffices"
public static final java.lang.String	OFFICE_LOCATIONS_FILE_SETTING	"officeLocationsFile"
public static final java.lang.String	OFFICE_MAX_WAIT_TIME_SETTING	"officeMaxWaitTime"
public static final java.lang.String	OFFICE_MIN_WAIT_TIME_SETTING	"officeMinWaitTime"
public static final java.lang.String	OFFICE_SIZE_SETTING	"officeSize"
public static final java.lang.String	OFFICE_WAIT_TIME_PARETO_COEFF_SETTING	"officeWaitTimeParetoCoeff"
public static final java.lang.String	WORK_DAY_LENGTH_SETTING	"workDayLength"

movement.StationaryMovement

public static final java.lang.String	LOCATION_S	"nodeLocation"
--------------------------------------	----------------------------	----------------

movement.WorkingDayMovement

public static final java.lang.String	PROBABILITY_TO_GO_SHOPPING_SETTING	"probGoShoppingAfterWork"
public static final java.lang.String	PROBABILITY_TO OWN CAR_SETTING	"ownCarProb"

movement.map.***movement.map.MapNode**

public static final int	MAX_TYPE	31
public static final int	MIN_TYPE	1

movement.map.MapRoute

public static final int	CIRCULAR	1
public static final int	PINGPONG	2

movement.map.PointsOfInterest

public static final java.lang.String	POI_FILE_S	"poiFile"
public static final java.lang.String	POI_NS	"PointsOfInterest"
public static final java.lang.String	POI_SELECT_S	"pois"

report.***report.AdjacencyGraphvizReport**

public static final java.lang.String	GRAPH_NAME	"adjgraph"
--------------------------------------	----------------------------	------------

report.ContactTimesReport

public static final java.lang.String	GRANULARITY	"granularity"
--------------------------------------	-----------------------------	---------------

report.DistanceDelayReport

public static final java.lang.String	SYNTAX	"distance at msg send, delivery time, hop count, MSG_ID"
--------------------------------------	------------------------	--

report.EnergyLevelReport

public static final java.lang.String	GRANULARITY	"granularity"
public static final java.lang.String	REPORTED_NODES	"nodes"

report.EventLogReport

public static final java.lang.String	MESSAGE_TRANS_DELIVERED	"D"
public static final java.lang.String	MESSAGE_TRANS_DELIVERED AGAIN	"A"
public static final java.lang.String	MESSAGE_TRANS_RELAYED	"R"

report.MessageDelayReport

public static final java.lang.String	HEADER	"# messageDelay cumulativeProbability"
--------------------------------------	------------------------	--

report.MessageGraphvizReport

public static final java.lang.String	GRAPH_NAME	"msggraph"
--------------------------------------	----------------------------	------------

report.MessageLocationReport

public static final java.lang.String	GRANULARITY	"granularity"
public static final java.lang.String	REPORTED_MESSAGES	"messages"

report.MessageReport

public static final java.lang.String	HEADER	"# messages: ID, start time, end time"
--------------------------------------	------------------------	--

report.MovementNs2Report

public static final java.lang.String	COORD_FORMAT	"%.5f"
public static final java.lang.String	DEF_NODE_ARRAY	"\$node_"
public static final java.lang.String	DEF_NS_CMD	"\$ns_"
public static final double	EPSILON	1.0E-5
public static final java.lang.String	NODE_ARR_S	"nodeArray"
public static final java.lang.String	NS_CMD_S	"nsCmd"

report.Report

public static final int	DEF_PRECISION	4
public static final java.lang.String	INTERVAL_SETTING	"interval"
public static final java.lang.String	INTERVALLED_FORMAT	"%04d.txt"
public static final java.lang.String	NAN	"NaN"
public static final java.lang.String	OUT_SUFFIX	".txt"
public static final java.lang.String	OUTPUT_SETTING	"output"
public static final java.lang.String	PRECISION_SETTING	"precision"
public static final java.lang.String	REPORT_NS	"Report"
public static final java.lang.String	REPORTDIR_SETTING	"Report.reportDir"
public static final java.lang.String	WARMUP_S	"warmup"

report.TotalContactTimeReport

public static final java.lang.String	HEADER	"# time totalContactTime"
--------------------------------------	------------------------	---------------------------

routing.***routing.ActiveRouter**

public static final java.lang.String	DELETE_DELIVERED_S	"deleteDelivered"
public static final java.lang.String	RESPONSE_PREFIX	"R_"

routing.EnergyAwareRouter

public static final java.lang.String	ENERGY_VALUE_ID	"Energy.value"
public static final java.lang.String	INIT_ENERGY_S	"intialEnergy"
public static final java.lang.String	SCAN_ENERGY_S	"scanEnergy"
public static final java.lang.String	TRANSMIT_ENERGY_S	"transmitEnergy"

public static final java.lang.String WARMUP_S	"energyWarmup"
---	----------------

routing.MaxPropRouter

public static final java.lang.String ALPHA_S	"alpha"
public static final double DEFAULT_ALPHA	1.0
public static final int DEFAULT_PROB_SET_MAX_SIZE	50
public static final java.lang.String MAXPROP_NS	"MaxPropRouter"
public static final java.lang.String PROB_SET_MAX_SIZE_S	"probSetMaxSize"

routing.MaxPropRouterWithEstimation

public static final double DEFAULT_ALPHA	1.0
public static final java.lang.String MAXPROP_NS	"MaxPropRouterWithEstimation"
public static final java.lang.String TIME_SCALE_S	"timeScale"

routing.MessageRouter

public static final java.lang.String B_SIZE_S	"bufferSize"
public static final int DENIED_NO_SPACE	-2
public static final int DENIED_OLD	-1
public static final int DENIED_TTL	-3
public static final int DENIED_UNSPECIFIED	-999
public static final java.lang.String MSG_TTL_S	"msgTtl"
public static final int Q_MODE_FIFO	2
public static final int Q_MODE_RANDOM	1
public static final int RCV_OK	0
public static final java.lang.String SEND_QUEUE_MODE_S	"sendQueue"
public static final int TRY_LATER_BUSY	1

routing.ProphetRouter

public static final java.lang.String BETA_S	"beta"
public static final double DEFAULT_BETA	0.25
public static final double GAMMA	0.98
public static final double P_INIT	0.75
public static final java.lang.String PROPHET_NS	"ProphetRouter"
public static final java.lang.String SECONDS_IN_UNIT_S	"secondsInTimeUnit"

routing.ProphetRouterWithEstimation

public static final java.lang.String BETA_S	"beta"
public static final double DEFAULT_BETA	0.25
public static final double DEFAULT_PTARGET	0.2
public static final double GAMMA	0.98
public static final java.lang.String P_AVG_TARGET_S	"targetPavg"
public static final double P_INIT	0.75
public static final java.lang.String PROPHET_NS	"ProphetRouterWithEstimation"
public static final java.lang.String TIME_SCALE_S	"timeScale"

routing.SprayAndWaitRouter

public static final java.lang.String BINARY_MODE	"binaryMode"
public static final java.lang.String MSG_COUNT_PROPERTY	"SprayAndWaitRouter.copies"
public static final java.lang.String NROF_COPIES	"nrofCopies"
public static final java.lang.String SPRAYANDWAIT_NS	"SprayAndWaitRouter"

routing.maxprop.***routing.maxprop.MeetingProbabilitySet**

```
public static final int INFINITE\_SET\_SIZE 2147483647
```

ui.*

[ui.DTNSimTextUI](#)

```
public static final long UI\_UP\_INTERVAL 60000L
```

[ui.DTNSimUI](#)

public static final java.lang.String	MM_WARMUP_S	"MovementModel.warmup"
public static final java.lang.String	NROF_REPORT_S	"Report.nrofReports"
public static final java.lang.String	REPORT_S	"Report.report"

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Hierarchy For Package routing

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
- routing.[MessageRouter](#)
 - routing.[ActiveRouter](#)
 - routing.[DirectDeliveryRouter](#)
 - routing.[EnergyAwareRouter](#) (implements core.[ModuleCommunicationListener](#))
 - routing.[EpidemicOracleRouter](#)
 - routing.[EpidemicRouter](#)
 - routing.[FirstContactRouter](#)
 - routing.[MaxPropRouter](#)
 - routing.[MaxPropRouterWithEstimation](#)
 - routing.[ProphetRouter](#)
 - routing.[ProphetRouterWithEstimation](#)
 - routing.[SprayAndWaitRouter](#)
 - routing.[PassiveRouter](#)
 - routing.[RoutingInfo](#)

Hierarchy For Package report

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - report.[ContactTimesReport.ConnectionInfo](#)
 - report.[Report](#)
 - report.[AdjacencyGraphvizReport](#) (implements core.[ConnectionListener](#))
 - report.[ConnectivityDtnsim2Report](#) (implements core.[ConnectionListener](#))
 - report.[ConnectivityONEReport](#) (implements core.[ConnectionListener](#))
 - report.[ContactsDuringAnICTReport](#) (implements core.[ConnectionListener](#), core.[UpdateListener](#))
 - report.[ContactsPerHourReport](#) (implements core.[ConnectionListener](#))
 - report.[ContactTimesReport](#) (implements core.[ConnectionListener](#))
 - report.[InterContactTimesReport](#)
 - report.[TotalContactTimeReport](#) (implements core.[UpdateListener](#))
 - report.[CreatedMessagesReport](#) (implements core.[MessageListener](#))
 - report.[DeliveredMessagesReport](#) (implements core.[MessageListener](#))
 - report.[DistanceDelayReport](#) (implements core.[MessageListener](#))
 - report.[DTN2Reporter](#) (implements core.[MessageListener](#))
 - report.[EncountersVSUniqueEncountersReport](#) (implements core.[ConnectionListener](#), core.[UpdateListener](#))
 - report.[EnergyLevelReport](#) (implements core.[UpdateListener](#))
 - report.[EventLogReport](#) (implements core.[ConnectionListener](#), core.[MessageListener](#))
 - report.[MessageDelayReport](#) (implements core.[MessageListener](#))
 - report.[MessageDeliveryReport](#) (implements core.[MessageListener](#))
 - report.[MessageGraphvizReport](#) (implements core.[MessageListener](#))
 - report.[MessageLocationReport](#) (implements core.[UpdateListener](#))
 - report.[MessageReport](#) (implements core.[MessageListener](#))
 - report.[MessageStatsReport](#) (implements core.[MessageListener](#))
 - report.[MovementNs2Report](#) (implements core.[MovementListener](#))
 - report.[PingAppReporter](#) (implements core.[ApplicationListener](#))
 - report.[TotalEncountersReport](#) (implements core.[ConnectionListener](#), core.[UpdateListener](#))
 - report.[UniqueEncountersReport](#) (implements core.[ConnectionListener](#), core.[UpdateListener](#))

Hierarchy For Package core

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - core.[Application](#)
 - core.[Connection](#)
 - core.[CBRConnection](#)
 - core.[VBRConnection](#)
 - core.[Coord](#) (implements java.lang.Cloneable, java.lang.Comparable<T>)
 - core.[Debug](#)
 - core.[DTN2Manager](#)
 - core.[DTN2Manager.EIDHost](#)
 - core.[DTNHost](#) (implements java.lang.Comparable<T>)
 - core.[DTNSim](#)
 - core.[Message](#) (implements java.lang.Comparable<T>)
 - core.[ModuleCommunicationBus](#)
 - core.[NetworkInterface](#) (implements core.[ModuleCommunicationListener](#))
 - core.[ParetoRNG](#)
 - core.[Settings](#)
 - core.[SimClock](#)
 - core.[SimScenario](#) (implements java.io.Serializable)
 - java.lang.Throwable (implements java.io.Serializable)
 - java.lang.Error
 - java.lang.AssertionError
 - core.[SimError](#)
 - core.[SettingsError](#)
 - core.[Tuple](#)<K,V>
 - core.[World](#)

Interface Hierarchy

- core.[ApplicationListener](#)
- core.[ConnectionListener](#)
- core.[MessageListener](#)
- core.[ModuleCommunicationListener](#)
- core.[MovementListener](#)
- core.[UpdateListener](#)

Hierarchy For Package input

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - input.[BinaryEventsReader](#) (implements input.[ExternalEventsReader](#))
 - input.[DTN2Events](#) (implements input.[EventQueue](#))
 - input.[DTN2Events.ParserHandler](#)
 - input.[EventQueueHandler](#)
 - input.[ExternalEvent](#) (implements java.lang.Comparable<T>, java.io.Serializable)
 - input.[ConnectionEvent](#)
 - input.[MessageEvent](#)
 - input.[MessageCreateEvent](#)
 - input.[MessageDeleteEvent](#)
 - input.[MessageRelayEvent](#)
 - input.[ExternalEventsQueue](#) (implements input.[EventQueue](#))
 - input.[ExternalMovementReader](#)
 - input.[MessageEventGenerator](#) (implements input.[EventQueue](#))
 - input.[MessageBurstGenerator](#)
 - input.[OneFromEachMessageGenerator](#)
 - input.[OneToEachMessageGenerator](#)
 - input.[ScheduledUpdatesQueue](#) (implements input.[EventQueue](#))
 - input.[StandardEventsReader](#) (implements input.[ExternalEventsReader](#))
 - input.[WKTReader](#)
 - input.[WKTMapReader](#)

Interface Hierarchy

- input.[EventQueue](#)
- input.[ExternalEventsReader](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)PREV [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Hierarchy For Package applications

Package Hierarchies:[All Packages](#)

Class Hierarchy

- [java.lang.Object](#)
 - [core.Application](#)
 - [applications.PingApplication](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)PREV [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Hierarchy For Package gui

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- [java.lang.Object](#)
 - [java.awt.Component](#) (implements [java.awt.image.ImageObserver](#), [java.awt.MenuContainer](#), [java.io.Serializable](#))
 - [java.awt.Container](#)
 - [javax.swing.JComponent](#) (implements [java.io.Serializable](#))
 - [javax.swing.JMenuBar](#) (implements [javax.accessibility.Accessible](#), [javax.swing.MenuElement](#))
 - [gui.SimMenuBar](#) (implements [java.awt.event.ActionListener](#))
 - [javax.swing.JPanel](#) (implements [javax.accessibility.Accessible](#))
 - [gui.EventLogControlPanel](#) (implements [java.awt.event.ActionListener](#))
 - [gui.EventLogPanel](#) (implements [java.awt.event.ActionListener](#), [core.ConnectionListener](#), [core.MessageListener](#))
 - [gui.GUIControls](#) (implements [java.awt.event.ActionListener](#), [javax.swing.event.ChangeListener](#))
 - [gui.InfoPanel](#) (implements [java.awt.event.ActionListener](#))
 - [gui.NodeChooser](#) (implements [java.awt.event.ActionListener](#))
 - [java.awt.Window](#) (implements [javax.accessibility.Accessible](#))
 - [java.awt.Frame](#) (implements [java.awt.MenuContainer](#))
 - [javax.swing.JFrame](#) (implements [javax.accessibility.Accessible](#), [javax.swing.RootPaneContainer](#), [javax.swing.WindowConstants](#))
 - [gui.MainWindow](#)
 - [gui.RoutingInfoWindow](#) (implements [java.awt.event.ActionListener](#))
 - [ui.DTNSimUI](#)
 - [gui.DTNSimGUI](#)
 - [gui.EventLogControl](#)

Hierarchy For Package gui.playfield

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- [java.lang.Object](#)
 - [java.awt.Component](#) (implements [java.awt.image.ImageObserver](#), [java.awt.MenuContainer](#), [java.io.Serializable](#))
 - [java.awt.Container](#)
 - [javax.swing.JComponent](#) (implements [java.io.Serializable](#))
 - [javax.swing.JPanel](#) (implements [javax.accessibility.Accessible](#))
 - [gui.playfield.PlayField](#)
 - [gui.playfield.PlayFieldGraphic](#)
 - [gui.playfield.MapGraphic](#)
 - [gui.playfield.MessageGraphic](#)
 - [gui.playfield.NodeGraphic](#)
 - [gui.playfield.PathGraphic](#)
 - [gui.playfield.ScaleReferenceGraphic](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Hierarchy For Package interfaces

Package Hierarchies:[All Packages](#)

Class Hierarchy

- java.lang.Object
 - interfaces.[ConnectivityGrid.GridCell](#)
 - interfaces.[ConnectivityOptimizer](#)
 - interfaces.[ConnectivityGrid](#)
 - core.[NetworkInterface](#) (implements core.[ModuleCommunicationListener](#))
 - interfaces.[InterferenceLimitedInterface](#)
 - interfaces.[SimpleBroadcastInterface](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Hierarchy For Package movement.map

Package Hierarchies:[All Packages](#)

Class Hierarchy

- java.lang.Object
 - movement.map.[DijkstraPathFinder](#)
 - movement.map.[MapNode](#) (implements java.lang.Comparable<T>)
 - movement.map.[MapRoute](#)
 - movement.map.[PointsOfInterest](#)
 - movement.map.[SimMap](#) (implements java.io.Serializable)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Hierarchy For Package routing.maxprop

Package Hierarchies:[All Packages](#)

Class Hierarchy

- java.lang.Object
 - routing.maxprop [MaxPropDijkstra](#)
 - routing.maxprop [MeetingProbabilitySet](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Hierarchy For Package routing.schedule

Package Hierarchies:[All Packages](#)

Class Hierarchy

- java.lang.Object
 - routing.schedule.[ScheduleDijkstra](#)
 - routing.schedule.[ScheduleEntry](#) (implements java.io.Serializable)
 - routing.schedule.[ScheduleOracle](#) (implements java.io.Serializable)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

Hierarchy For Package ui

Package Hierarchies:[All Packages](#)

Class Hierarchy

- [java.lang.Object](#)
 - [ui.DTNSimUI](#)
 - [ui.DTNSimTextUI](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | CONSTR | [METHOD](#)

interfaces

Class ConnectivityGrid.GridCell

```
java.lang.Object
└─interfaces.ConnectivityGrid.GridCell
```

Enclosing class:[ConnectivityGrid](#)

```
public class ConnectivityGrid.GridCell
extends java.lang.Object
```

A single cell in the cell grid. Contains the interfaces that are currently in that part of the grid.

Method Summary

void	addInterface(NetworkInterface ni) Adds an interface to this cell
java.util.ArrayList<NetworkInterface>	getInterfaces() Returns a list of of interfaces in this cell
void	moveInterface(NetworkInterface ni, ConnectivityGrid.GridCell to) Moves a interface in a Cell to another Cell
void	removeInterface(NetworkInterface ni) Removes an interface from this cell
java.lang.String	toString() Returns a string representation of the cell

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Method Detail

getInterfaces

```
public java.util.ArrayList<NetworkInterface> getInterfaces()
```

Returns a list of of interfaces in this cell

Returns:

a list of of interfaces in this cell

addInterface

```
public void addInterface(NetworkInterface ni)
```

Adds an interface to this cell

Parameters:

ni - The interface to add

removeInterface

```
public void removeInterface(NetworkInterface ni)
```

Removes an interface from this cell

Parameters:

ni - The interface to remove

moveInterface

```
public void moveInterface(NetworkInterface ni,  
                         ConnectivityGrid.GridCell to)
```

Moves a interface in a Cell to another Cell

Parameters:

ni - The interface to move
to - The cell where the interface should be moved to

toString

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

Returns a string representation of the cell

Overrides:

`toString` in class `java.lang.Object`

Returns:

a string representation of the cell

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | CONSTR | [METHOD](#)

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV CLASS](#) [NEXT CLASS](#)SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

report

Class ContactTimesReport.ConnectionInfo

```
java.lang.Object
└ report.ContactTimesReport.ConnectionInfo
```

Enclosing class:[ContactTimesReport](#)

```
protected class ContactTimesReport.ConnectionInfo
extends java.lang.Object
```

Objects of this class store time information about contacts.

Constructor Summary

[ContactTimesReport.ConnectionInfo\(DTNHost h1, DTNHost h2\)](#)

Method Summary

void	connectionEnd() Should be called when the connection ended to record the time.
boolean	equals(java.lang.Object other) Returns true if the other connection info contains the same hosts.
double	getConnectionTime() Returns the time that passed between creation of this info and call to connectionEnd() .
int	hashCode() Returns the same hash for ConnectionInfos that have the same two hosts.
java.lang.String	toString() Returns a string representation of the info object

Methods inherited from class java.lang.Object

clone, finalize, getClass, notify, notifyAll, wait, wait, wait

Constructor Detail

ContactTimesReport.ConnectionInfo

```
public ContactTimesReport.ConnectionInfo(DTNHost h1,
                                         DTNHost h2)
```

Method Detail

connectionEnd

```
public void connectionEnd()
```

Should be called when the connection ended to record the time. Otherwise [getConnectionTime\(\)](#) will use end time as the time of the request.

getConnectionTime

```
public double getConnectionTime()
```

Returns the time that passed between creation of this info and call to [connectionEnd\(\)](#). Unless connectionEnd() is called, the difference between start time and current sim time is returned.

Returns:

The amount of simulated seconds passed between creation of this info and calling connectionEnd()

equals

```
public boolean equals(java.lang.Object other)
```

Returns true if the other connection info contains the same hosts.

Overrides:

`equals` in class `java.lang.Object`

hashCode

```
public int hashCode()
```

Returns the same hash for ConnectionInfos that have the same two hosts.

Overrides:

`hashCode` in class `java.lang.Object`

Returns:

Hash code

toString

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

Returns a string representation of the info object

Overrides:

`toString` in class `java.lang.Object`

Returns:

a string representation of the info object

input

Class DTN2Events.ParserHandler

```
java.lang.Object
└─ input.DTN2Events.ParserHandler
```

Enclosing class:

[DTN2Events](#)

```
public class DTN2Events.ParserHandler
extends java.lang.Object
```

Inner class that implements the CLA interface for receiving bundles from dtnd.

Constructor Summary

DTN2Events.ParserHandler (int hostID, DTN2Events eventsHandler, java.lang.String consoleHost, int consolePort)	Creates a new parser handler.
--	-------------------------------

Method Summary

void	connected()
boolean	error (java.lang.String reason, java.lang.Exception exception, boolean fatal)
BundleTransferReceipt	incomingBundle (java.lang.String location, CLAParser.BundleAttributes attributes)
boolean	parseError (java.lang.String reason)

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

DTN2Events.ParserHandler

```
public DTN2Events.ParserHandler(int hostID,
                               DTN2Events eventsHandler,
                               java.lang.String consoleHost,
                               int consolePort)
```

Creates a new parser handler.

Parameters:

hostID - ID of the host that this parser corresponds to
eventsHandler - Reference to the events handler
consoleHost - Hostname of the dtnd
consolePort - Console port of the dtnd

Method Detail**incomingBundle**

```
public BundleTransferReceipt incomingBundle(java.lang.String location,
CLAParser.BundleAttributes attributes)
```

connected

```
public void connected()
```

error

```
public boolean error(java.lang.String reason,
java.lang.Exception exception,
boolean fatal)
```

parseError

```
public boolean parseError(java.lang.String reason)
```

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

Serialized Form

Package core

Class [core.SettingsError](#) extends [SimError](#) implements Serializable

Class [core.SimError](#) extends [java.lang.AssertionError](#) implements Serializable

Serialized Fields

e

`java.lang.Exception e`

Class [core.SimScenario](#) extends [java.lang.Object](#) implements Serializable

Serialized Fields

world

[World](#) world

The world instance

hosts

`java.util.List<E> hosts`

List of hosts in this simulation

name

`java.lang.String name`

Name of the simulation

nrofGroups

`int nrofGroups`

number of host groups

worldSizeX

`int worldsizeX`

Width of the world

worldSizeY

`int worldsizeY`

Height of the world

maxHostRange

`double maxHostRange`

Largest host's radio range

endTime

`double endTime`

Simulation end time

updateInterval

`double updateInterval`

Update interval of sim time

eqHandler

`EventQueueHandler eqHandler`

External events queue

simulateConnections

`boolean simulateConnections`

Should connections between hosts be simulated

simMap

`SimMap simMap`

Map used for host movement (if any)

connectionListeners

```
java.util.List<E> connectionListeners
```

Global connection event listeners

messageListeners

```
java.util.List<E> messageListeners
```

Global message event listeners

movementListeners

```
java.util.List<E> movementListeners
```

Global movement event listeners

updateListeners

```
java.util.List<E> updateListeners
```

Global update event listeners

appListeners

```
java.util.List<E> appListeners
```

Global application event listeners

Package gui

**Class [gui.EventLogControlPanel](#) extends javax.swing.JPanel
implements Serializable**

Serialized Fields

smallFont

```
java.awt.Font smallFont
```

headingFont

```
java.awt.Font headingFont
```

logControls

Serialized Form

```
java.util.Vector<E> logControls
```

showAllCheck

```
javax.swing.JCheckBox showAllCheck
```

pauseAllCheck

```
javax.swing.JCheckBox pauseAllCheck
```

layout

```
java.awt.GridBagLayout layout
```

c

```
java.awt.GridBagConstraints c
```

Class [gui.EventLogPanel](#) extends javax.swing.JPanel implements Serializable

Serialized Fields

regExp

```
java.lang.String regExp
```

Regular expression to filter log entries (changed through Settings)

maxNrofEvents

```
int maxNrofEvents
```

how many events to show in log (changed through Settings)

font

```
java.awt.Font font
```

gui

```
DTNSimGUI gui
```

eventPanes

```
java.util.Vector<E> eventPanes
```

layout

```
java.awt.GridLayout layout
```

controls

```
EventLogControlPanel controls
```

conUpCheck

```
EventLogControl conUpCheck
```

conDownCheck

```
EventLogControl conDownCheck
```

msgCreateCheck

```
EventLogControl msgCreateCheck
```

msgTransferStartCheck

```
EventLogControl msgTransferStartCheck
```

msgRelayCheck

```
EventLogControl msgRelayCheck
```

msgRemoveCheck

```
EventLogControl msgRemoveCheck
```

msgDeliveredCheck

```
EventLogControl msgDeliveredCheck
```

msgDropCheck

```
EventLogControl msgDropCheck
```

msgAbortCheck

```
EventLogControl msgAbortCheck
```

Class [gui.GUIControls](#) extends javax.swing.JPanel implements Serializable

Serialized Fields

simTimeField

```
javax.swing.JTextField simTimeField
```

sepsField

```
javax.swing.JLabel sepsField
```

playButton

```
javax.swing.JButton playButton
```

playUntilButton

```
javax.swing.JButton playUntilButton
```

paused

```
boolean paused
```

stepButton

```
javax.swing.JButton stepButton
```

step

```
boolean step
```

ffwButton

```
javax.swing.JButton ffwButton
```

isFFw

```
boolean isFFw
```

oldSpeedIndex

```
int oldSpeedIndex
```

screenShotButton

```
javax.swing.JButton screenShotButton
```

guiUpdateChooser

```
javax.swing.JComboBox guiUpdateChooser
```

guiUpdateInterval

```
double guiUpdateInterval
```

zoomSelector

```
javax.swing.JSpinner zoomSelector
```

pf

[PlayField](#) pf

gui

[DTNSimGUI](#) gui

lastUpdate

```
long lastUpdate
```

lastSimTime

```
double lastSimTime
```

playUntilTime

```
double playUntilTime
```

useHourDisplay

```
boolean useHourDisplay
```

Class [gui.InfoPanel](#) extends javax.swing.JPanel implements Serializable

Serialized Fields

msgChooser

```
javax.swing.JComboBox msgChooser
```

info

```
javax.swing.JLabel info
```

infoButton

```
javax.swing.JButton infoButton
```

routingInfoButton

```
javax.swing.JButton routingInfoButton
```

selectedMessage

```
Message selectedMessage
```

selectedHost

```
DTNHost selectedHost
```

gui

```
DTNSimGUI gui
```

Class [gui.MainWindow](#) extends javax.swing.JFrame implements Serializable

Serialized Fields

playFieldScroll

```
javax.swing.JScrollPane playFieldScroll
```

Class [gui.NodeChooser](#) extends javax.swing.JPanel implements Serializable

Serialized Fields

gui

```
DTNSimGUI gui
```

nodes

```
java.util.List<E> nodes
```

groupChooser

```
javax.swing.JComboBox groupChooser
```

nodesPanel

```
javax.swing.JPanel nodesPanel
```

chooserPanel

```
javax.swing.JPanel chooserPanel
```

Class [gui.RoutingInfoWindow](#) extends javax.swing.JFrame implements Serializable

Serialized Fields

host

```
DTNHost host
```

refreshButton

```
javax.swing.JButton refreshButton
```

treePane

```
javax.swing.JScrollPane treePane
```

Class [gui.SimMenuBar](#) extends javax.swing.JMenuBar implements Serializable

Serialized Fields

enableBgImage

```
javax.swing.JCheckBoxMenuItem enableBgImage
```

enableNodeName

```
javax.swing.JCheckBoxMenuItem enableNodeName
```

enableNodeCoverage

```
javax.swing.JCheckBoxMenuItem enableNodeCoverage
```

enableNodeConnections

```
javax.swing.JCheckBoxMenuItem enableNodeConnections
```

enableMapGraphic

```
javax.swing.JCheckBoxMenuItem enableMapGraphic
```

autoClearOverlay

```
javax.swing.JCheckBoxMenuItem autoClearOverlay
```

clearOverlay

```
javax.swing.JMenuItem clearOverlay
```

about

```
javax.swing.JMenuItem about
```

field

[PlayField](#) field

Package gui.playfield

Class [gui.playfield.PlayField](#) extends javax.swing.JPanel implements Serializable

Serialized Fields

w

[World](#) w

bgColor

```
java.awt.Color bgColor
```

overlayGraphics

```
java.util.List<E> overlayGraphics
```

autoClearOverlay

Serialized Form

```
boolean autoClearOverlay
```

mapGraphic

```
MapGraphic mapGraphic
```

showMapGraphic

```
boolean showMapGraphic
```

refGraphic

```
ScaleReferenceGraphic refGraphic
```

underlayImage

```
java.awt.image.BufferedImage underlayImage
```

imageTransform

```
java.awt.geom.AffineTransform imageTransform
```

curTransform

```
java.awt.geom.AffineTransform curTransform
```

underlayImgDx

```
double underlayImgDx
```

underlayImgDy

```
double underlayImgDy
```

Package input

Class [input.ConnectionEvent](#) extends [ExternalEvent](#) implements
Serializable

Serialized Fields

fromAddr

```
int fromAddr
```

address of the node the (dis)connection is from

toAddr

int **toAddr**

address of the node the (dis)connection is to

isUp

boolean **isUp**

Is this a "connection up" event

interfaceId

java.lang.String **interfaceId**

What is the interface number for this event

Class [input.ExternalEvent](#) extends java.lang.Object implements Serializable

Serialized Fields

time

double **time**

Time of the event (simulated seconds)

Class [input.MessageCreateEvent](#) extends [MessageEvent](#) implements Serializable

Serialized Fields

size

int **size**

responseSize

int **responseSize**

Class [input.MessageDeleteEvent](#) extends [MessageEvent](#) implements Serializable

Serialized Fields

drop

`boolean drop`

is the delete caused by a drop (not "normal" removing)

Class [input.MessageEvent](#) extends [ExternalEvent](#) implements Serializable

Serialized Fields

fromAddr

`int fromAddr`

address of the node the message is from

toAddr

`int toAddr`

address of the node the message is to

id

`java.lang.String id`

identifier of the message

Class [input.MessageRelayEvent](#) extends [MessageEvent](#) implements Serializable

Serialized Fields

stage

`int stage`

Package movement.map

Class [movement.map.SimMap](#) extends `java.lang.Object` implements Serializable

Serialized Fields

minBound`Coord minBound`**maxBound**`Coord maxBound`**nodes**`java.util.ArrayList<E> nodes`

list representation of the map for efficient list-returning

nodesMap`java.util.Map<K,V> nodesMap`

hash map presentation of the map for efficient finding node by coord

offset`Coord offset`

offset of map translations

isMirrored`boolean isMirrored`

is this map data mirrored after reading

needsRehash`boolean needsRehash`

is re-hash needed before using hash mode (some coordinates changed)

Package routing.schedule**Class [routing.schedule.ScheduleEntry](#) extends java.lang.Object
implements Serializable**

serialVersionUID: 42L

Serialized Fields

time

```
double time
```

from

```
int from
```

to

```
int to
```

via

```
int via
```

delta

```
double delta
```

duration

```
double duration
```

usageCount

```
int usageCount
```

**Class [routing.schedule.ScheduleOracle](#) extends [java.lang.Object](#)
implements Serializable**

serialVersionUID: 42L

Serialized Fields

schedules

```
java.util.Map<K,V> schedules
```

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)