User’s Guide

MultiCastor 2.0

V 1.0 (beta2)

DHBW Stuttgart
Duale Hochschule
Baden-Württemberg

6. Mai 2012

Project: MultiCastor 2.0

Client:
Rentschler & Stuckert
Rotebühlplatz 41/1
70178 Stuttgart

Contractor:
TIT10AID - Team 4 - MCastor2.0

Fabian Fäßler
Filip Haase
Matthis Hauschild
Sebastian Koralewski
Jonas Traub
Christopher Westphal

Rotebühlplatz 41 – Raum 0.10
70178 Stuttgart

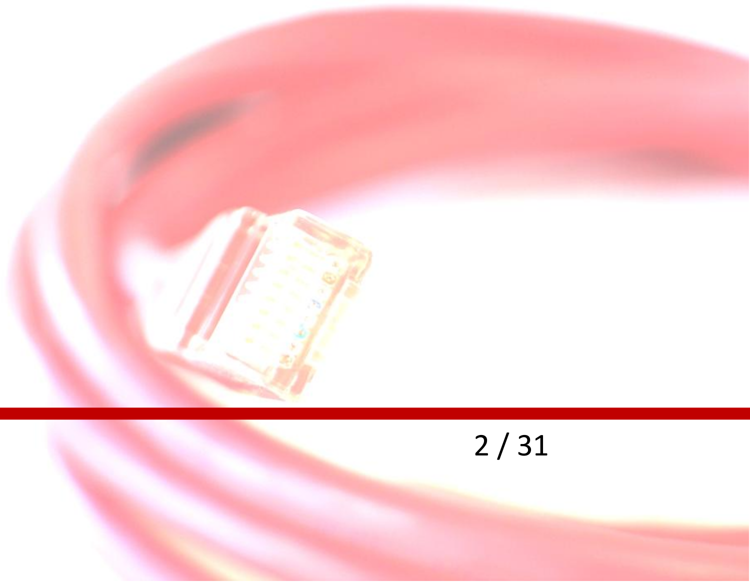


Table of contents

1. About MultiCastor2.0

MultiCastor2.0 is an open source project, developed by students of the DHBW Stuttgart and published under GPLv3.

The software is able to do an extensive multicast network test. It supports both Layer 3 (IPv4 and IPv6) and Layer 2 (via MMRP path registration).

If you would like to test the multicast ability of your network hardware, the MultiCastor is a great tool to do so. You can download it here: http://sourceforge.net/projects/multicastor/ and use it without charge.

2. Functions of the MultiCastor

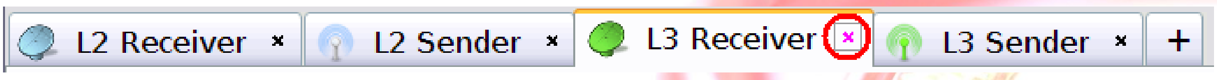
2.1. GUI Functions

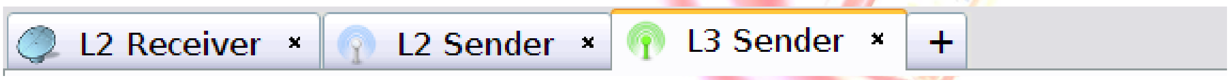
2.1.1. Switch, Disable and Enable Tabs

The management of multicast senders and receivers is realized with a tabbing system. There are four different tabs:

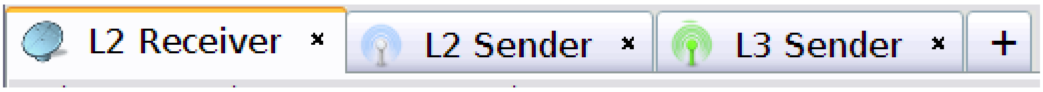
- Layer 2 Sender
- Layer 2 Receiver
- Layer 3 Sender
- Layer 3 Receiver

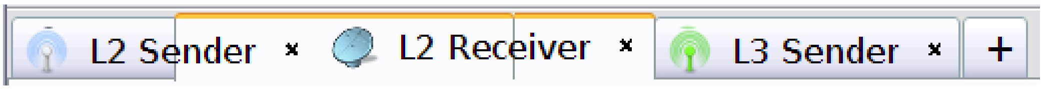
If you do not need every tab, you can disable the unused by clicking the x-sybmol in the respective tab.

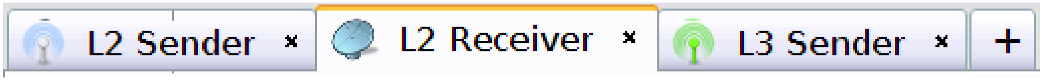




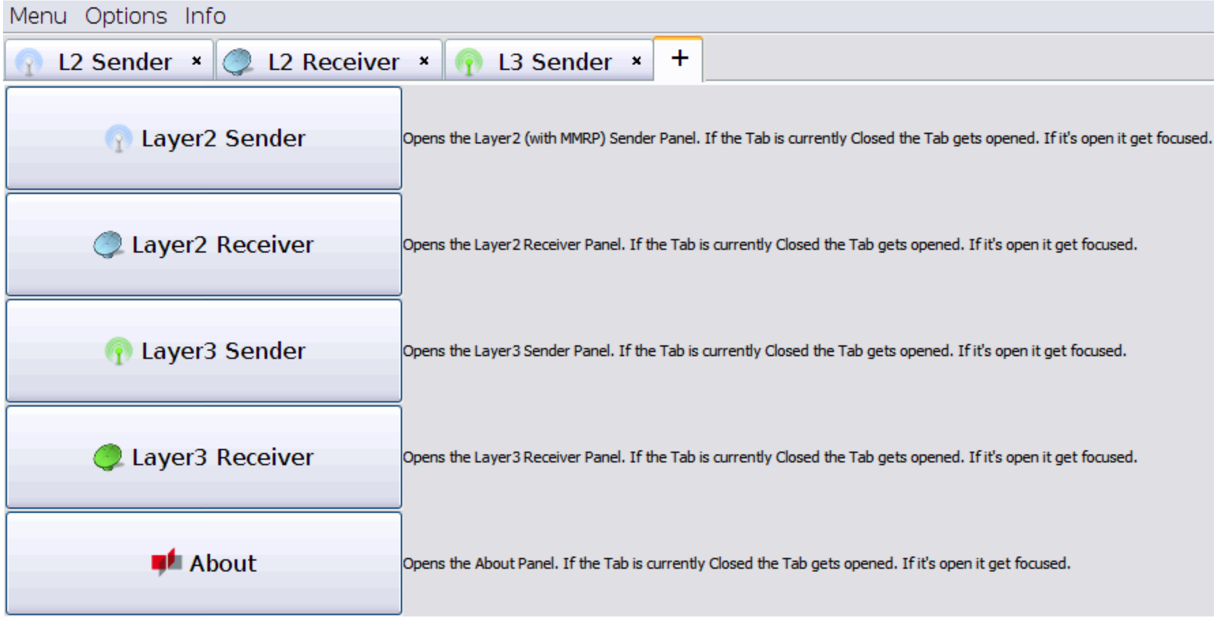
You can readjust the remaining tabs by your wishes.







To reactivate the disabled L3 Receiver tab, click on the + symbol at the right corner of the tab bar.



Opens the Layer2 (with MMRP) Sender Panel. If the Tab is currently Closed the Tab gets opened. If it's open it get focused.

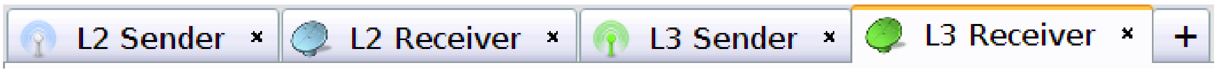
Opens the Layer2 Receiver Panel. If the Tab is currently Closed the Tab gets opened. If it's open it get focused.

Opens the Layer3 Sender Panel. If the Tab is currently Closed the Tab gets opened. If it's open it get focused.

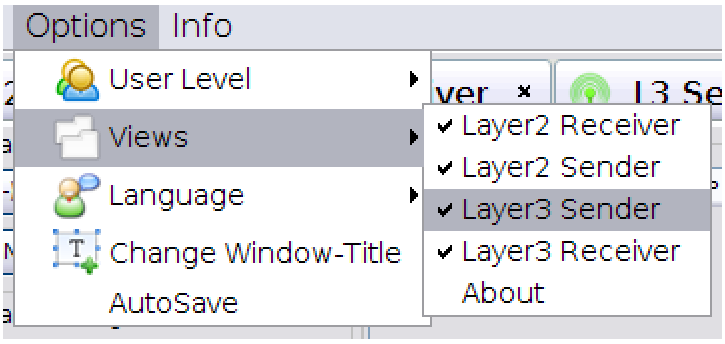
Opens the Layer3 Receiver Panel. If the Tab is currently Closed the Tab gets opened. If it's open it get focused.

Opens the About Panel. If the Tab is currently Closed the Tab gets opened. If it's open it get focused.

In this new view, you habe to click on the corresponding button to finally reactivate it

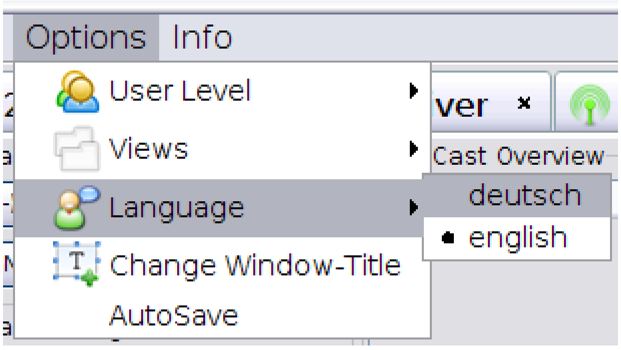


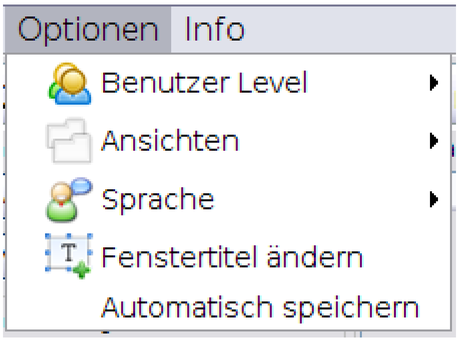
An alternative way to enable and disable tabs is to access them via the menu item Options and then Views.



2.1.2. Change Language Settings

The MultiCastor supports multiple languages. Each translation is in a separate languagefile. To choose another language, simply click on the menu item Options and then Language.





After selecting the desired langauge, every label in the MultiCastor changes according to the new language. You do not need to restart the programm.

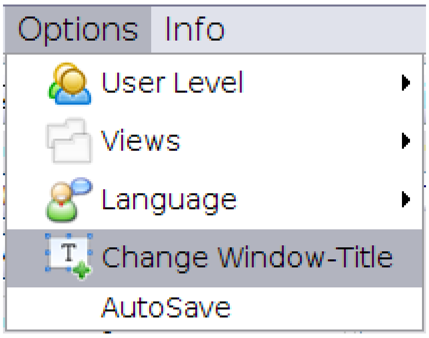
2.1.3. Change the Window Title

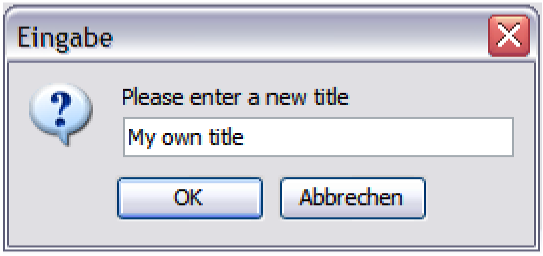
To facilitate the use of multiple instances of the MultiCastor on one computer, you can choose an individual instance title for each instance. This instance title will be used as the window title and as the programm name in the task bar.

This title can be devided into two parts.

First, the (optional) individually choosen title and second, the title of the currently shown tab. The default setting is „MultiCastor: <tabname>“.

You can change the first part via „Change Window-Title“ in the Options menu item.

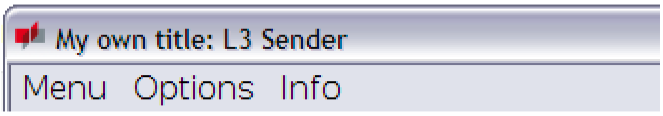



Eingabe

Please enter a new title

My own title

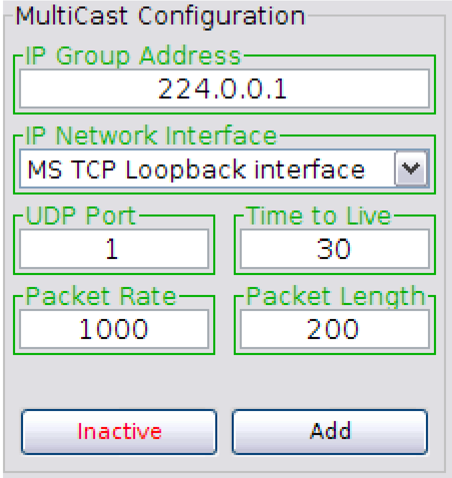
Choose an arbitrary title and click „OK“. Now you can see your changes in the window title:


My own title: L3 Sender

2.2. Create senders and receivers

To create a new sender, click on the button New as a start. Thereafter, specify your sender in the MultiCast Configuration panel.

In the following example, we will create a layer 3 IPv4 sender.


MultiCast Configuration

IP Group Address: 224.0.0.1

IP Network Interface: MS TCP Loopback interface

UDP Port: 1

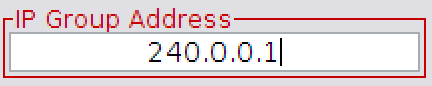
Time to Live: 30

Packet Rate: 1000

Packet Length: 200

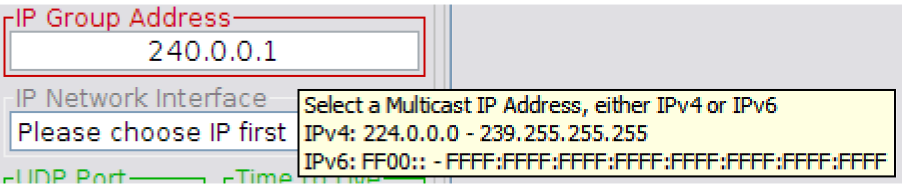
The first step is to specify a multicast IP address. Since this input is used to distinguish between an IPv4 and an IPv6 sender, the definition of the network interface and the packet length is only enabled if the IP address is valid.

This is indicated by a green border around the relevant input field. If you see a red border you have to correct your input data.



IP Group Address
240.0.0.1

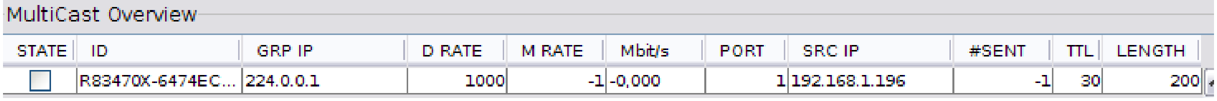
If you need to know the valid facet, take a look at the tooltips which are displayed when you move your cursor over the respective input field.



IP Group Address
240.0.0.1
IP Network Interface
Please choose IP first

Select a Multicast IP Address, either IPv4 or IPv6
IPv4: 224.0.0.0 - 239.255.255.255
IPv6: FF00:: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

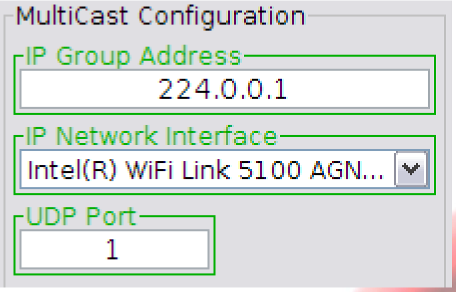
Lastly, click on the button „Add“ to create the sender. The sender appears in the table in the MultiCast Overview area.



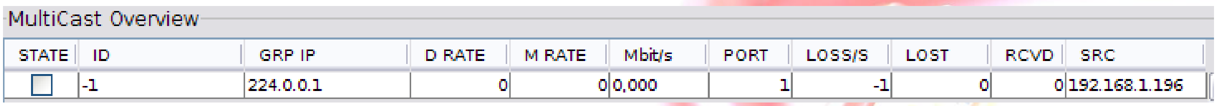
MultiCast Overview

STATE	ID	GRP IP	D RATE	M RATE	Mbit/s	PORT	SRC IP	#SENT	TTL	LENGTH
□	R83470X-6474EC...	224.0.0.1	1000	-1	-0,000	1	192.168.1.196	-1	30	200

To create a receiver, use the same procedure as to create a sender. That is why you fill in valid data first and then click on „Add“ to add the receiver to the table afterwards.



MultiCast Configuration
IP Group Address
224.0.0.1
IP Network Interface
Intel(R) WiFi Link 5100 AGN...
UDP Port
1



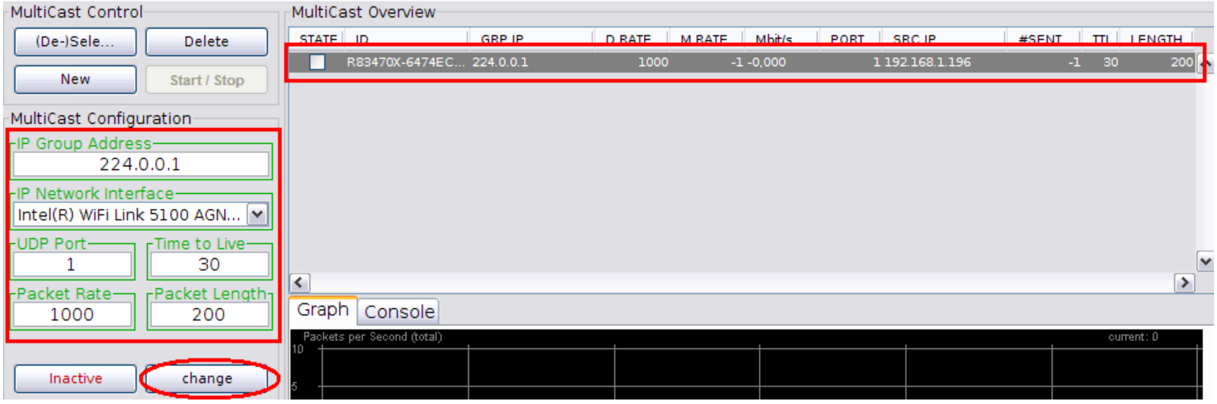
MultiCast Overview

STATE	ID	GRP IP	D RATE	M RATE	Mbit/s	PORT	LOSS/S	LOST	RCVD	SRC
□	-1	224.0.0.1	0	0	0,000	1	-1	0	0	192.168.1.196

2.3. Change Senders and Receivers

To change already existing senders and receivers, there are two possibilities.

To just change one data record, select it by clicking on it in the table. The data associated with this record appears in the MultiCast Configuration panel. Change it to fit your needs and click on the button „change" to apply the data.



MultiCast Configuration

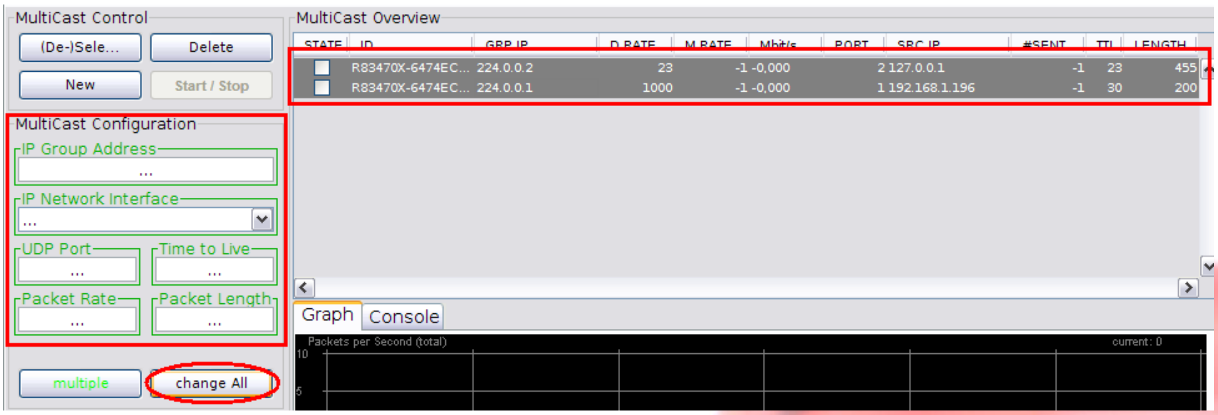
- IP Group Address: 224.0.0.1
- IP Network Interface: Intel(R) WiFi Link 5100 AGN...
- UDP Port: 1
- Time to Live: 30
- Packet Rate: 1000
- Packet Length: 200

MultiCast Overview

STATE	ID	GRP IP	D RATE	M RATE	Mbit/s	PORT	SRC IP	#SENT	TTL	LENGTH
□	R83470X-6474EC...	224.0.0.1	1000	-1	-0,000	1	192.168.1.196	-1	30	200

To change more than one record at a time, choose multiple senders/receivers by clicking on them with pressed ctrl- or shift-key. This time you see „..." in each input field. Do your changes but leave the three dots in the input fields you do not want to change.

By selecting „change All", you apply your changes to all selected datasets.



MultiCast Configuration

- IP Group Address: ...
- IP Network Interface: ...
- UDP Port: ...
- Time to Live: ...
- Packet Rate: ...
- Packet Length: ...

MultiCast Overview

STATE	ID	GRP IP	D RATE	M RATE	Mbit/s	PORT	SRC IP	#SENT	TTL	LENGTH
□	R83470X-6474EC...	224.0.0.2	23	-1	-0,000	2	127.0.0.1	-1	23	455
□	R83470X-6474EC...	224.0.0.1	1000	-1	-0,000	1	192.168.1.196	-1	30	200

Via the „active/inactive/multiple“ button in the lower left-hand corner, you can change the status of activity of your multicast data records. The changes take effect by clicking on the „change“ and „change All“ button respectively.

2.4. Start/Stop Senders and Receivers

To start a sender or receiver, simply activate the checkbox in the state column of the table. If you do so, the sender/receiver immediately starts its service.

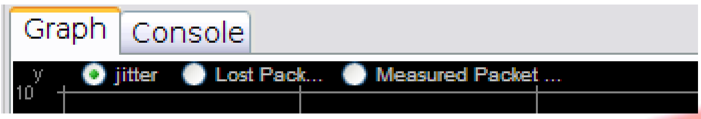
Beyond that, you can start multiple datasets by selecting them with your ctrl- or shift-key, switching the button „multiple“ to „active“ and confirming that by clicking on „change All“.

2.5. Tracing a network test

2.5.1. Graph View

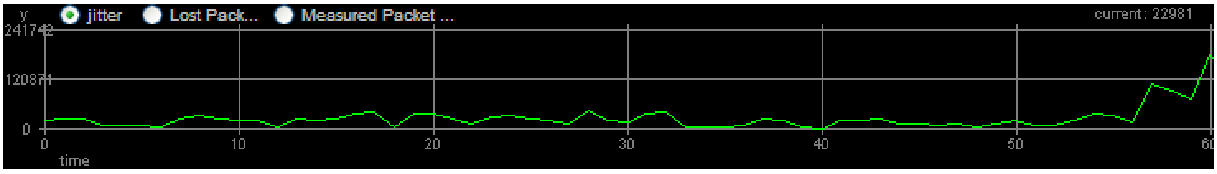
With support of the graph view, you can trace the different parameters on the receiving side.

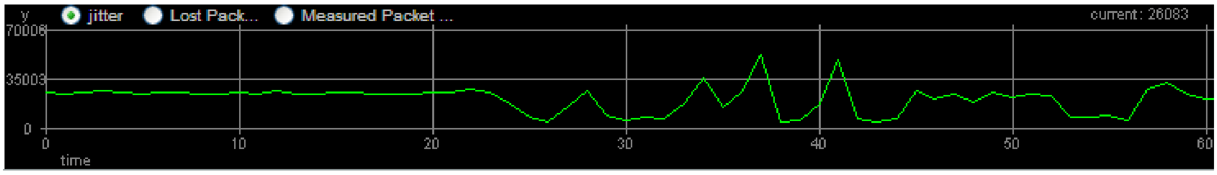
Choose the property to trace by selecting the corresponding radio button.



Jitter

The jitter shows the clock deviation during transfer of packets. Even if the sender sends with a constant clock, there can be jitter due to a different load of switches and computers within your testing network. More information on jitter can be found on Wikipedia[1] for example.





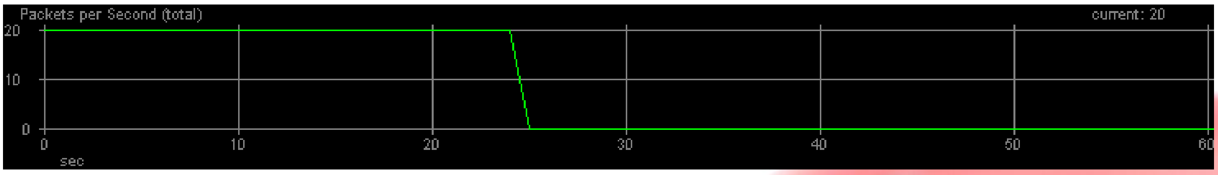
Lost Packet Rate

This graph shows the sum of lost packets over time. The MultiCastor sends an increasing number with each packet, with which the receiver can calculate the number of lost packets.

Measured Packet Rate

This graph shows the measured packet rate over time.

In a sender tab, this graph always shows the number of packets per second of **all** senders.

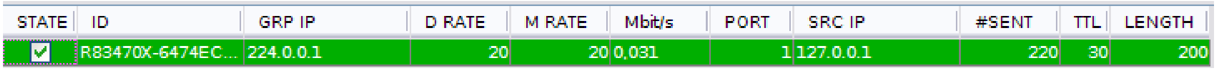


This graphic shows the result of a loopback test in which one sender sends 20 packets per second for 25 seconds.

[1] http://en.wikipedia.org/wiki/Jitter

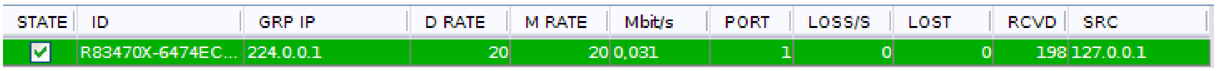
2.5.2. Current Data in the Table

In addition to the graph view, you can use the table and trace each dataset separately. The refresh intervall is one second.



STATE	ID	GRP IP	D RATE	M RATE	Mbit/s	PORT	SRC IP	#SENT	TTL	LENGTH
☑	R83470X-6474EC...	224.0.0.1	20	20	0,031	1	127.0.0.1	220	30	200

Example of an active sender

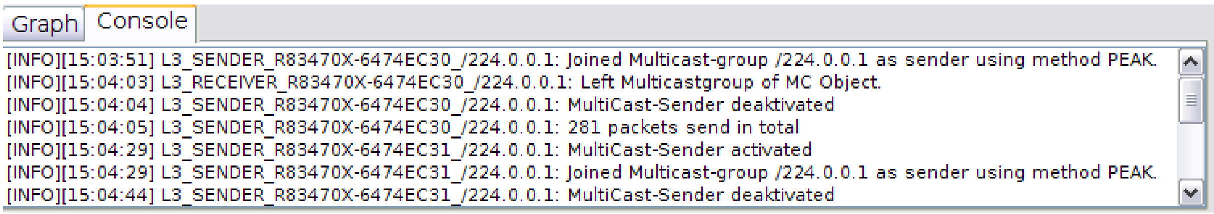


STATE	ID	GRP IP	D RATE	M RATE	Mbit/s	PORT	LOSS/S	LOST	RCVD	SRC
☑	R83470X-6474EC...	224.0.0.1	20	20	0,031	1	0	0	198	127.0.0.1

Example of an active receiver

2.5.3. Console Output

As soon as you activate a sender or receiver, a status is printed out to the program's internal console as well as to the system's console.



```
[INFO][15:03:51] L3_SENDER_R83470X-6474EC30_/224.0.0.1: Joined Multicast-group /224.0.0.1 as sender using method PEAK.
[INFO][15:04:03] L3_RECEIVER_R83470X-6474EC30_/224.0.0.1: Left Multicastgroup of MC Object.
[INFO][15:04:04] L3_SENDER_R83470X-6474EC30_/224.0.0.1: MultiCast-Sender deaktivated
[INFO][15:04:05] L3_SENDER_R83470X-6474EC30_/224.0.0.1: 281 packets send in total
[INFO][15:04:29] L3_SENDER_R83470X-6474EC31_/224.0.0.1: MultiCast-Sender activated
[INFO][15:04:29] L3_SENDER_R83470X-6474EC31_/224.0.0.1: Joined Multicast-group /224.0.0.1 as sender using method PEAK.
[INFO][15:04:44] L3_SENDER_R83470X-6474EC31_/224.0.0.1: MultiCast-Sender deaktivated
```

```
[INFO][15:04:04] L3_SENDER_R83470X-6474EC30_/224.0.0.1: MultiCast-Sender deaktivated
[INFO][15:04:05] L3_SENDER_R83470X-6474EC30_/224.0.0.1: 281 packets send in total
[INFO][15:04:29] L3_SENDER_R83470X-6474EC31_/224.0.0.1: MultiCast-Sender activated
[INFO][15:04:29] L3_SENDER_R83470X-6474EC31_/224.0.0.1: Joined Multicast-group /224.0.0.1
as sender using method PEAK.
[INFO][15:04:44] L3_SENDER_R83470X-6474EC31_/224.0.0.1: MultiCast-Sender deaktivated
[INFO][15:04:45] L3_SENDER_R83470X-6474EC31_/224.0.0.1: 321 packets send in total
[INFO][15:07:04] L3_RECEIVER_R83470X-6474EC30_/224.0.0.1: Started Receiver with MC
Object.
[INFO][15:07:06] L3_SENDER_R83470X-6474EC32_/224.0.0.1: MultiCast-Sender activated
[INFO][15:07:06] L3_SENDER_R83470X-6474EC32_/224.0.0.1: Joined Multicast-group /224.0.0.1
as sender using method PEAK.
```

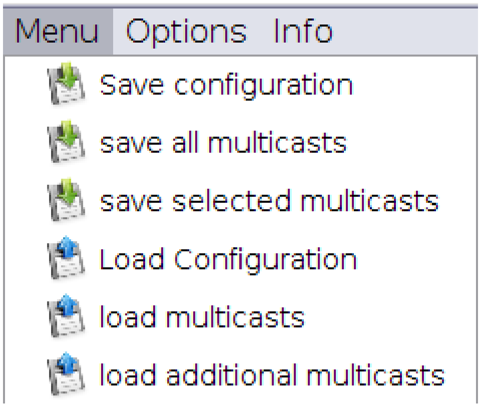
```
[INFO][15:07:29] L3_RECEIVER_R83470X-6474EC32_/224.0.0.1: Left Multicastgroup of MC
Object.
[INFO][15:07:30] L3_SENDER_R83470X-6474EC32_/224.0.0.1: MultiCast-Sender deaktivated
[INFO][15:07:31] L3_SENDER_R83470X-6474EC32_/224.0.0.1: 501 packets send in total
```

2.6. Autosave

If you activate the „autosave“ function via the „Options“ menu item, every time you change the GUI or multicast senders/receivers, the related configuration file is saved.

2.7. Sava and Load Configurations

The MultiCastor lets you choose between multiple possibilities to save and load different configurations. All of these functions are available in the „Menu“ menu item.



There are two different types of configuration files. First, there is the GUI configuration file which includes states of GUI elements like the selected language or the order of the tabs. Second, there is a configuration file for all multicast datasets.

This distinction between GUI and data helps when you share your multicast configurations with your colleagues but do not want to use their language settings for example.

2.7.1. Save Configuration

This functions saves your current GUI configuration.

The MultiCastor automatically saves your GUI configuration when you close it and restores it when you start it again, so you only need to save your GUI configuration, when you want to share it or use it on another computer.

2.7.2. Load Configuration

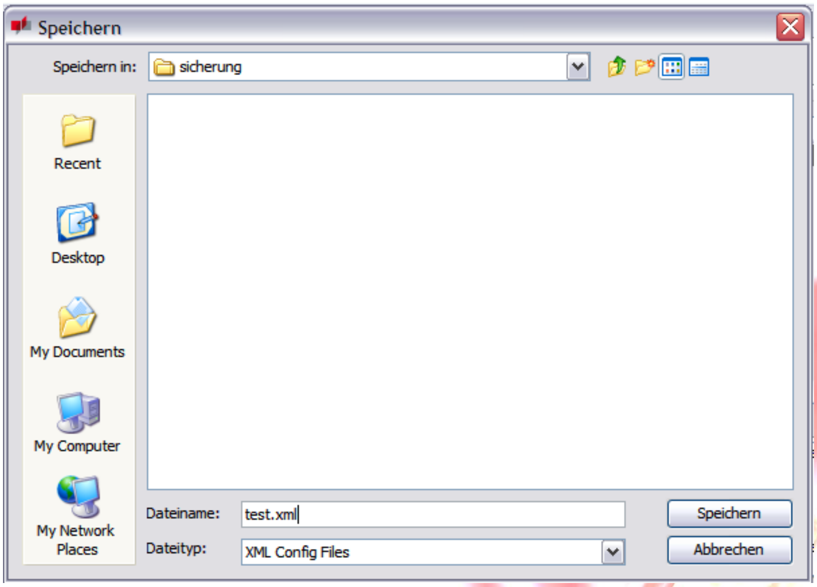
This functions loads an afore saved GUI configuration.

At program startup this is done automatically if there is a GUI configuration file called „GUIConfig.xml“ in the base directory of the MultiCastor

2.7.3. Save all Multicasts

This options allows you to save all multicast data records. The MultiCastor saves all layer 2 and layer 3 senders and receivers in one xml configuration file.

After selecting „Save all Multicasts“, you can choose a location and filename.



Speichern in: sicherung

Dateiname: test.xml

Dateityp: XML Config Files

2.7.4. Save selected Multicasts

There is one option to save only **selected** multicasts.

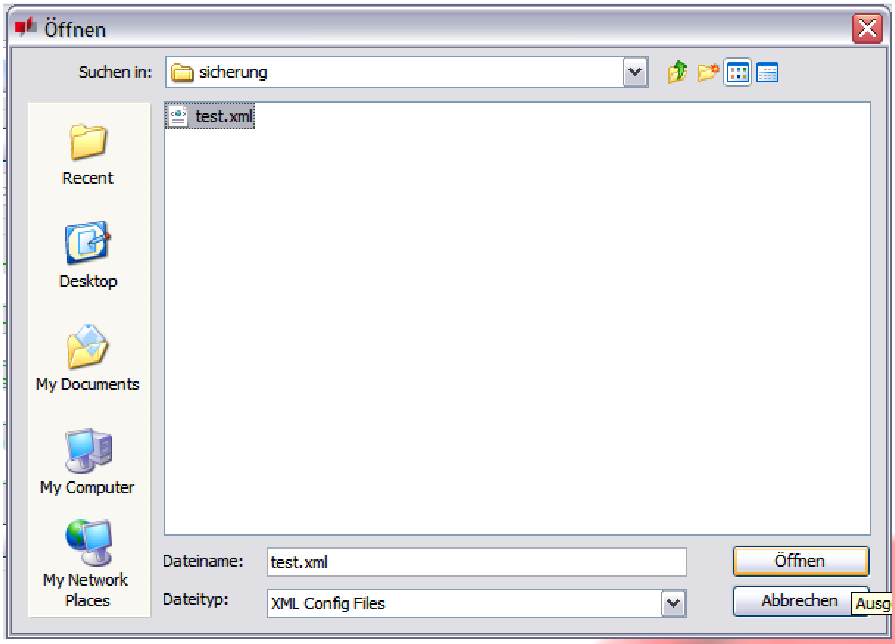
After clicking on „Save selected Multicasts“, you can specify location and filename like with saving all multicasts.

The difference is that only selected multicast data records will be saved. Therefore you should select the multicasts you want to save **over all tabs** before.

To select multiple datasets, use the ctrl- and shift-key.

2.7.5. Load Multicasts

This option loads a multicast configuration file.



Öffnen

Suchen in: sicherung

test.xml

Dateiname: test.xml

Dateityp: XML Config Files

Simply select a valid multicast configuration xml file and click „open“ to load it.

Please be aware of the following:

- This functions **replaces** all existing data records with the loaded ones
- The MultiCastor saves the activity status, too. That means if you load activated datasets, they begin to send/receive immediately after loading.

2.7.6. Load additional Multicasts

This function is used to load additional multicast datasets. The existing datasets will not be deleted, but the new will be appended to the according lists of multicasts.

Beyond that, this function behaves exactly like the simple loading of multicasts (see previous section)

3. Possible problems and their solutions

3.1. Not enough working memory

3.1.1. Less than 10 % of memory available

Description of cause

This message appears if less than 10% of your memory is available. This could especially arise if you have activated many multicasts with high packet rates.

Possible solutions

- Try to free your memory by closing other programs
- Deactivate senders/receivers or decrease their packet rate
- Do not use the autosave function of the MultiCastor
- Install new RAM or increase the memory useable by the Java Virtual Machine
- Start the MultiCastor without GUI

3.1.2. No memory available anymore

Description of cause

If there is no more memory left, the MultiCastor cannot work anymore and shuts down.

That can be caused due to:

- another program that uses a large amount of memory
- having not enough memory installed in your computer
- too less memory for the Java Virtual Machine.

Possible solutions

Try the possible solution points of the previos section, when there is just 10% of memory left.

If this message arises immediately after startup, try to provide the Java Virtual Machine with more memory.

3.2. Configurationfile not specified (at console use)

Description of cause

If you start the MultiCastor via commandline you need to specify a multicast configuration file. Otherwise you cannot start the program.

Possible solutions

Create a multicast configuration file and specify it as a parameter on the commandline. To create it, use the GUI or a XML editor.

3.3. Specified invalid parameter (at console use)

Description of cause

If you try to start the MultiCastor with invalid parameters on the commandline, you will get this message.

Possible solutions

Call the MultiCastor with the parameter **–h** in order to get a list with all valid parameters.

3.4. Could not create logfile

Description of cause

If the MultiCastor cannot create a logfile, there are two possible causes:

- You are not allowed to write in that directory
- There is not enough disk space to create the logfile

Possible solutions

You need write access in the MultiCastor directory. Either get that or change the directory to one where you have write access. Assure that there is enough disk space.

If you got the MultiCastor on a read-only medium (for example CD-ROM), copy it to your harddrive and start it from there.

You could start the MultiCastor without a logfile, too, but be aware that arising problems probably cannot be detected properly.

3.5. Refreshing of GUI takes rather long

Description of cause

This message arises if the refreshing of the graphical user interface takes longer than it should. This could be due to a heavy load on the system.

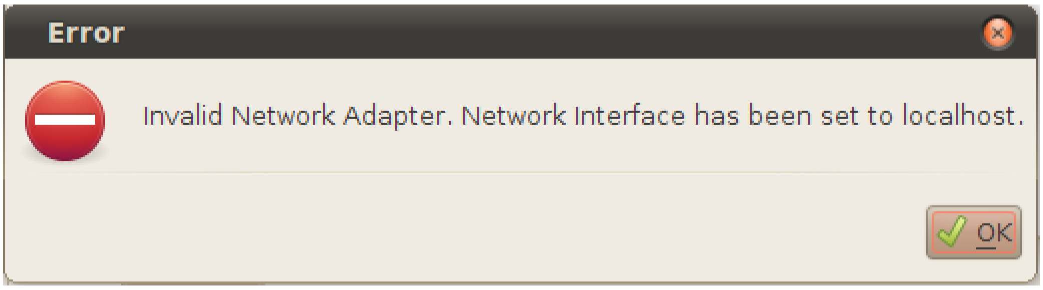
If the refreshing takes too long, the GUI will be difficult to controll

Possible solutions

Try to reduce the system load by:

- closing other programs with heavy load
- lowering the packet rate of senders or deactivate some senders

3.6. Invalid Network Adapter


Error

Invalid Network Adapter. Network Interface has been set to localhost.

Description of cause

This message arises if the specified adapter is not available. This can happen when you deactivate a network adapter after starting the MultiCastor or when an external network adapter has a power loss.

Especially if you save a multicast configuration on one computer and load it on another, the chance of a network adapter being not available is rather high.

Possible solutions

- Check whether your adapter is deactivated or indicates a failure.
- Check if the required drivers are installed correctly on your system.

If you cannot get the network adapter to work, you have to choose another one by changing the affected data record.

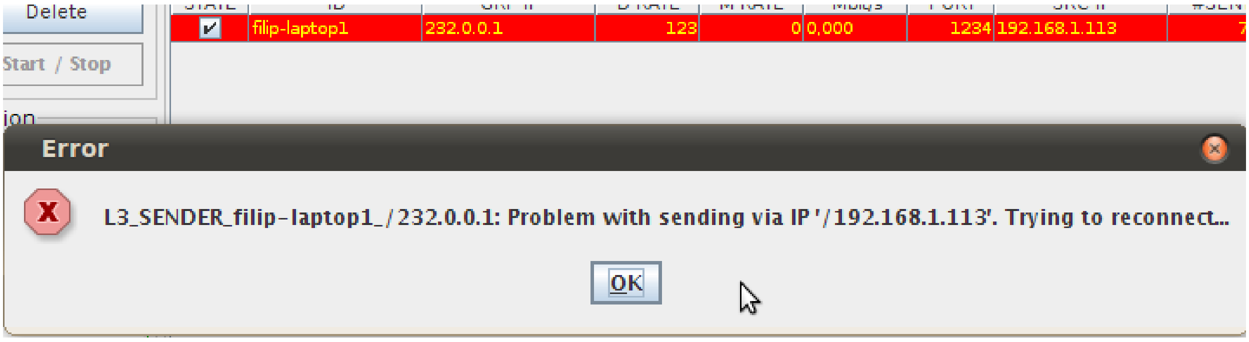
3.7. Loss of Network Connection

Description of cause

A loss of network connection can happen when someone plugs out the network cable or if your adapter has a power loss.

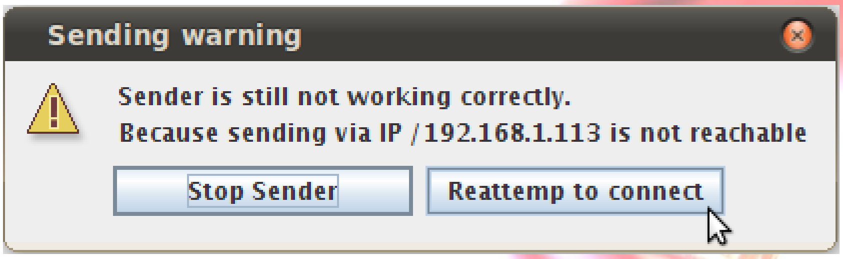
Possible solutions

The MultiCastor reacts to a loss of connection by displaying a message to the user and trying to reconnect.



L3_SENDER_filip-laptop1_/232.0.0.1: Problem with sending via IP '/192.168.1.113'. Trying to reconnect...

If a reconnect fails, you can choose to stop the sender or to reattempt connection for another 10 seconds.



Sender is still not working correctly.
Because sending via IP /192.168.1.113 is not reachable

4. Create and Use Own Language Files

4.1. Create an Own Language File

To save languages, the MultiCastor uses Java Property Files. The syntax of those files can be looked up at Wikipedia[2] for example.

In each language file, there are translations of a specific language, which are attributed to keys. During operation of the MultiCastor, the affected translations are loaded via the key from the concering language file.

Let us have a look to a small extract from the English language file.

```
miscBorder.ipGroupAddress        =IP Group Address
miscBorder.ipNetworkInterface    =IP Network Interface
miscBorder.MacGroupAddress       =Mac Group Adress
miscBorder.NetworkInterface      =Network Interface
config.message.ipFirst           =Please choose IP first
config.message.ipFirstShort      =IP first
```

In comparison to that, here is the same extract from the German language file, where the keys are identical and just the values have changed.

```
miscBorder.ipGroupAddress        =IP Gruppenadresse
miscBorder.ipNetworkInterface    =IP Netzwerk Interface
miscBorder.MacGroupAddress       =Mac Gruppenadresse
miscBorder.NetworkInterface      =Netzwerk Interface
config.message.ipFirst           =Bitte erst die IP wählen
config.message.ipFirstShort      =IP zuerst
```

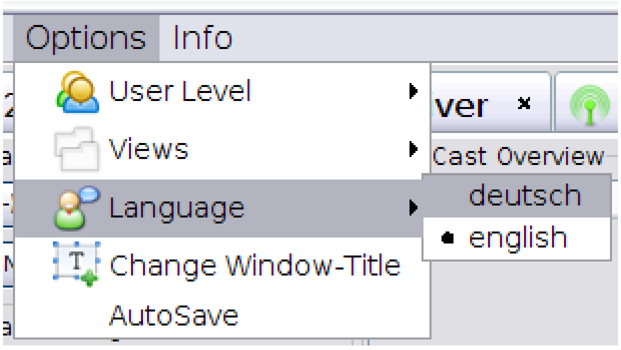
[2] http://en.wikipedia.org/wiki/.properties

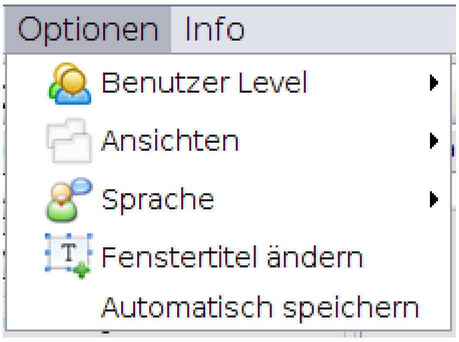
4.2. Use Own Language Files

Every language file is located in the subdirectory Language of the MultiCastor directory. To use a language file, which you created or received, simply copy it to that location and make sure you name it in the following format: „<LANGUAGENAME>.lang“

After restarting the MultiCastor, you can choose the new language through the

Options->Language menu item. Every text in the GUI should change to the new language immediately.





5. Connection to STAF / STAX

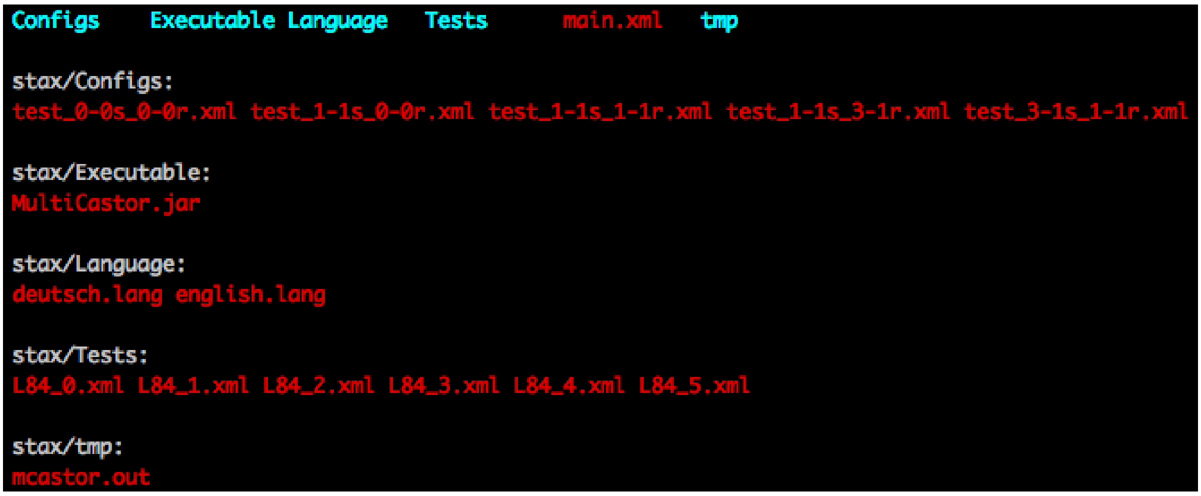
5.1. General Information

This chapter requires precognition about STAF/STAX. If you do not have previous knowledge, please read the section „About STAF/STAX“ in this document first.

To gain deeper knowledge about STAF/STAX, consider visiting the STAF Homepage (http://staf.sourceforge.net/)

5.2. Provided Files

We chose the following file structure for the provided tests:



```
Configs    Executable Language   Tests      main.xml   tmp

stax/Configs:
test_0-0s_0-0r.xml test_1-1s_0-0r.xml test_1-1s_1-1r.xml test_1-1s_3-1r.xml test_3-1s_1-1r.xml

stax/Executable:
MultiCastor.jar

stax/Language:
deutsch.lang english.lang

stax/Tests:
L84_0.xml L84_1.xml L84_2.xml L84_3.xml L84_4.xml L84_5.xml

stax/tmp:
mcastor.out
```

stax/main.xml is the testing script. It imports every test from **stax/Tests/*** and calls each of them. Furthermore, it provides functions for starting a MCastor2.0 instance for example.

stax/Configs/* contains the multicast configuration files to start senders and receivers. The chosen naming convention is the following:

Each configuration file starts with **test_**, followed by the number of senders and receivers:

test_*<total number of senders>***-***<number of senders to test>***s_***<total number of receivers>***-***<number of receivers to test>***.xml**

short:

test_*<1>***-***<2>***s_***<3>***-***<4>***.xml**

#	Name	Description
1	*<total number of senders>*	Number of senders, defined in the configfile
2	*<number of senders to test>*	Number of senders to check in the test
3	*<total number of receivers>*	Number of receivers, defined in the configfile
4	*<number of receivers to test>*	Number of receivers to check in the test

example: **test_3-1s_1-1r.xml**

There are three senders in total, from which one is relevant for the analysis. There is one receiver which is relevant for the analysis, too.

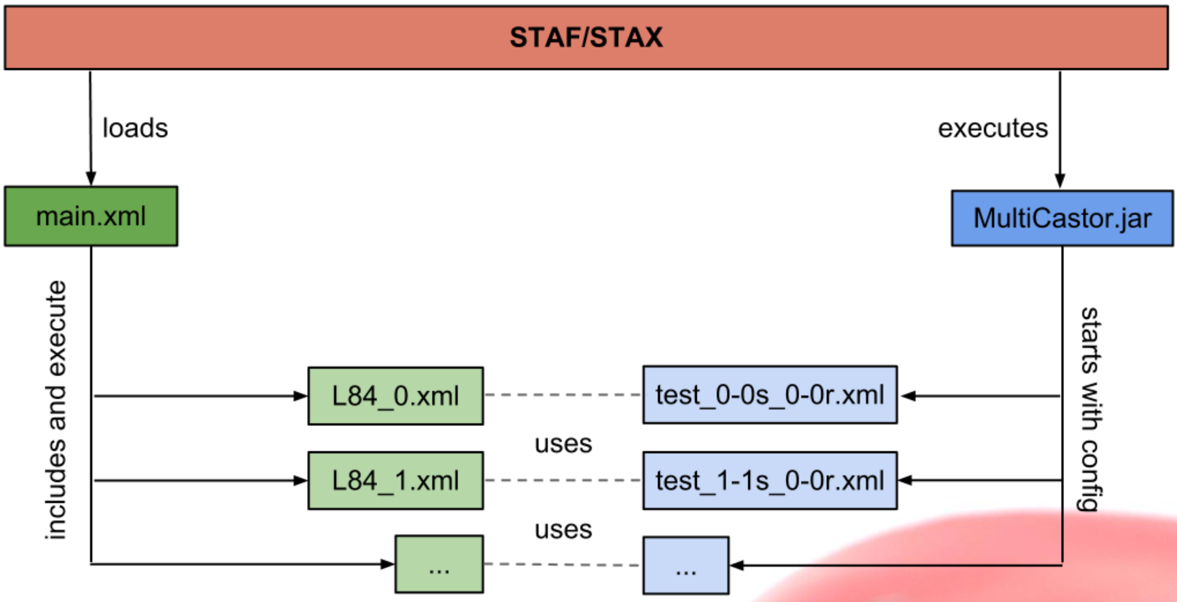
In **stax/Executable/**, you can find the **MultiCastor.jar** file used for the tests.

In **stax/Language** are the language files needed by the **MultiCastor.jar**. The provided tests only need the English language file **englisch.lang**.

In **stax/Tests/**, one can find the test scripts. Which script tests which functionality will be explained in detail later.

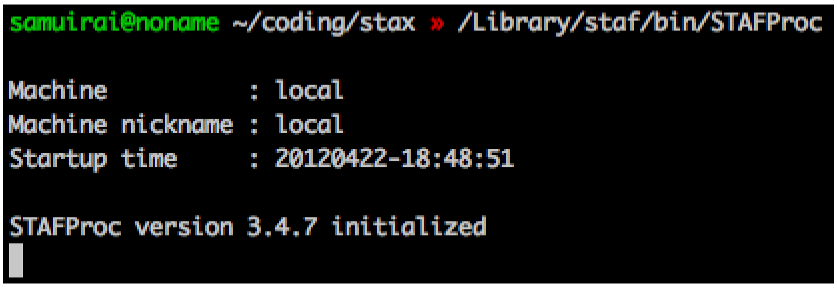
stax/tmp/mcastor.out is the file, where STDOUT (standart out) and STDERR (standard error) of the MultiCastor are piped to. This file is used for the analysis.

The following diagram briefly shows the connections between the mentioned files.



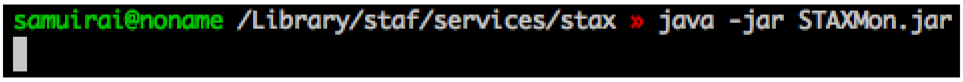
5.3. Execution of the Tests

To start the testing, you need to start a STAF Client (*STAFProc*) in the **stax/** subdirectory, because there are configuration files which are called relative to this path.



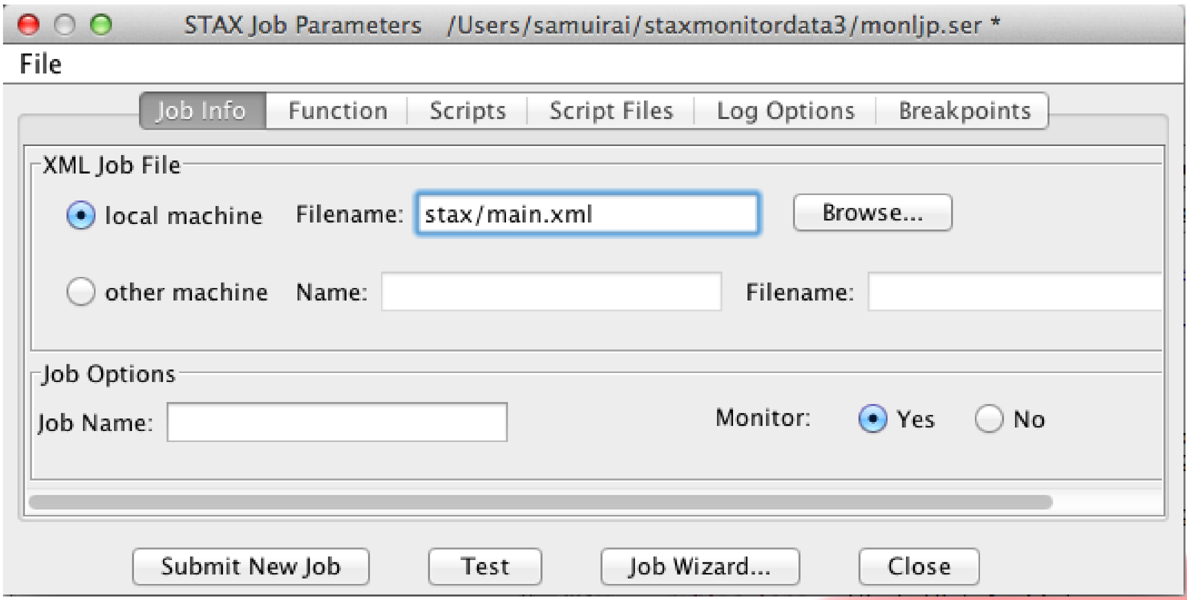
```
samuirai@noname ~/coding/stax » /Library/staf/bin/STAFProc

Machine          : local
Machine nickname : local
Startup time     : 20120422-18:48:51

STAFProc version 3.4.7 initialized
```

You need to start the STAX Monitor (*STAXMon*) subsequently.



```
samuirai@noname /Library/staf/services/stax » java -jar STAXMon.jar
```

Thereafter, you should select the main.xml via **File->Submit new Job**.



STAX Job Parameters /Users/samuirai/staxmonitordata3/monljp.ser *

XML Job File

● local machine Filename: stax/main.xml

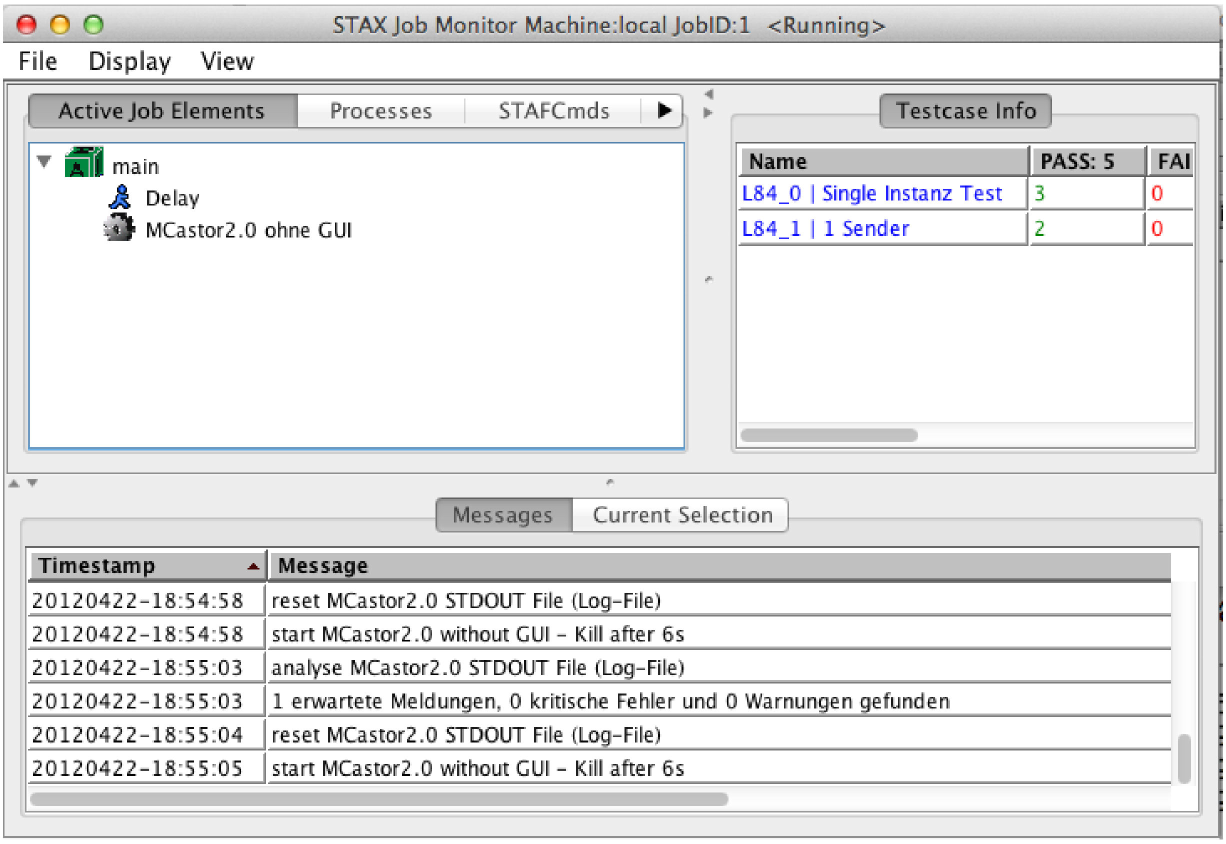
○ other machine Name: Filename:

Job Options

Job Name:

Monitor: ● Yes ○ No

Finally, click on **Submit New Job** to start the tests. The execution should approximately take one minute.



Name	PASS: 5	FAI
L84_0 | Single Instanz Test	3	0
L84_1 | 1 Sender	2	0

Timestamp	Message
20120422-18:54:58	reset MCastor2.0 STDOUT File (Log-File)
20120422-18:54:58	start MCastor2.0 without GUI – Kill after 6s
20120422-18:55:03	analyse MCastor2.0 STDOUT File (Log-File)
20120422-18:55:03	1 erwartete Meldungen, 0 kritische Fehler und 0 Warnungen gefunden
20120422-18:55:04	reset MCastor2.0 STDOUT File (Log-File)
20120422-18:55:05	start MCastor2.0 without GUI – Kill after 6s

6. About MMRP

The MMR protocoll is a layer 2 protocol and provides switches with the multicast ability. It was specified by the IEEE 802.1ak in 2007.

Via multicast a sender is able to reach multiple receivers without sending one packet multiple times. That is because the switch copies the packets and addresses all registered receivers.

First of all, a sender creates a multicast group. If a receiver wants to receive packets from this sender, it registers itself at the group. A receiver is always able to unsubscribe from a group, of course.

The main advantages of layer 2 multicasting are:

- Unload the network, because a sender needs to send a packet only once
- You do not need a router to use layer 2 multicast in comparison to layer 3 multicast.

Important to mention is, that you need switches, which support the MMR protocol in order to use it. At the time this manual was written, these switches are very rare.

7. About STAF / STAX

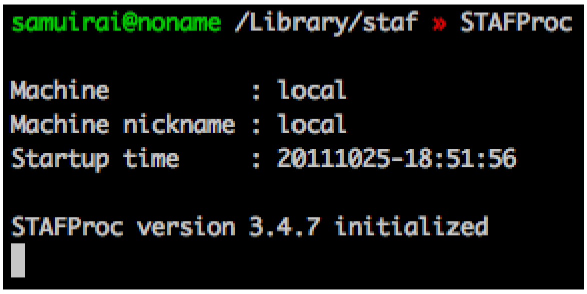
In this chapter one should get a overview what STAF/STAX is and how it works.

7.1. STAF

STAF is an **s**oftware **t**esting **a**utomation **f**ramework, which is able to execute tests on hundreds of computers at the same time. STAF provides many services, which can be used for testing efficently.

7.1.1. General:

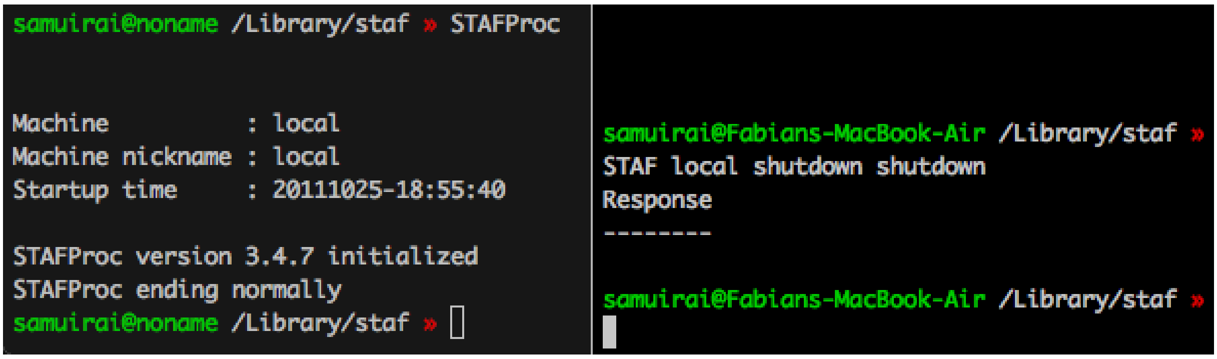
"STAFProc" starts a STAF server process.



```
samuirai@noname /Library/staf » STAFProc

Machine          : local
Machine nickname : local
Startup time     : 20111025-18:51:56

STAFProc version 3.4.7 initialized
```

To shut it down, use the command *„STAF local shutdown shutdown"* from another terminal session to avoid inconsistency, which may result from using Ctrl+C.

In the following examples, the STAF server will be on the left hand side and the STAF client on the right hand side.



```
samuirai@noname /Library/staf » STAFProc


Machine          : local
Machine nickname : local
Startup time     : 20111025-18:55:40

STAFProc version 3.4.7 initialized
STAFProc ending normally
samuirai@noname /Library/staf » ▯
```

```
samuirai@Fabians-MacBook-Air /Library/staf »
STAF local shutdown shutdown
Response
--------

samuirai@Fabians-MacBook-Air /Library/staf »
█
```

7.1.2. STAF Commands:

STAF <Endpoint> <Server> <Request>

<Endpoint> “local” if you want to execute the commands locally. Another computer name, if you want to execute them on a remote computer.

<Service> name of the service to be executed.

<Request> is the service request

“The STAF command line utility works just like any other STAF application. It registers with STAF, performs a request (which is the service request you specify), and then unregisters. That last step causes the handle to be deleted. This somewhat limits the usage of the STAF command line utility.“[3]

7.1.3. STAF Demo[4]:

In the provided STAF demonstration, you can estimate the power of STAF. You can see a STAF service written in java, which executes a simple java program.

In comparison to that, one can use STAX. The STAF execution engine uses a scripting language based on XML and python to let you easily create STAF services.

7.2.STAX

STAX (STAF eXecution Engine) is a STAF service to define tests with. It uses XML and Jython to create those tests and lets you start them via the STAF Monitor.

[3] Quelle: http://staf.sourceforge.net/current/STAFGS.pdf 4.3 Submitting STAF Request from Command Line

[4] http://staf.sourceforge.net/current/STAFGS.pdf 8. Getting started with STAF

8. History

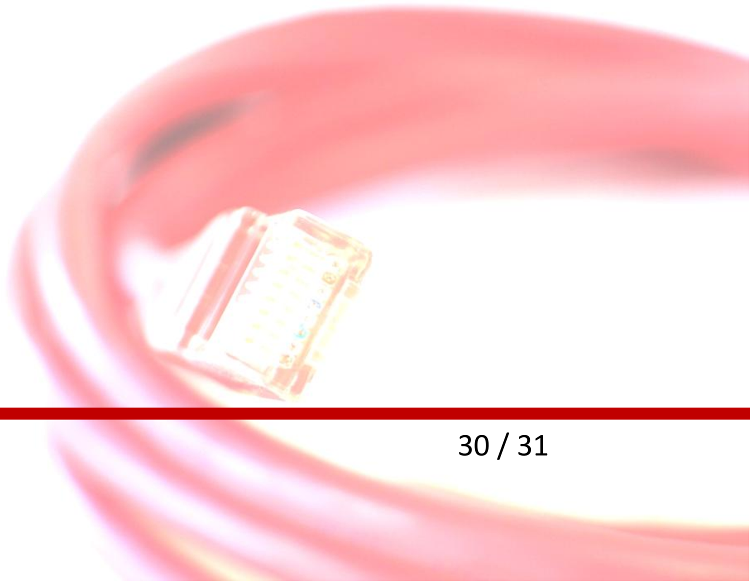
The MultiCastor is a project realised by students from the DHBW Stuttgart in Germany. The MultiCastor is published under the GPLv3 licence.

8.1. MultiCastor Version 1.0

- Bastian Wagener
- Johannes Beutel
- Thomas Lüder
- Daniel Becker
- Daniela Gerz
- Jannik Müller

8.2. MultiCastor Version 2.0

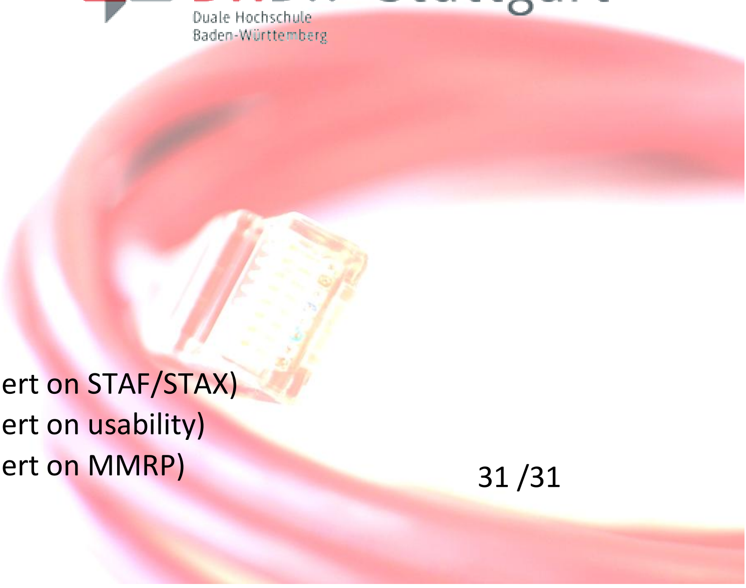
- Jonas Traub
- Matthis Hauschild
- Filip Haase
- Fabian Fäßler
- Sebastian Koralewski
- Christopher Westphal



Documentversions

Version	Date	Author	Comment
V 0.1	27.09.2011	Traub	create template
V 0.2	28.04.2012	Traub	create whole manual
V 1.0 (beta 1)	30.04.2012	Hauschild	conversion into this template
V 1.0 (beta 2)	05.05.2012	Hauschild	translation into English

DHBW Stuttgart
Duale Hochschule
Baden-Württemberg



Dokumentversion/Document Version:
Titel: User's Guide
Version: V 1.0 (beta2) (6. Mai 2012)

Autoren/Authors Projektteam/Project team:

- Jonas Traub (Projektleiter)
- Filip Haase (Leading Engineer)
- Matthis Hauschild (Documentation)
- Fabian Fäßler (Engineer/Tester, Expert on STAF/STAX)
- Christopher Westphal (Engineer/Tester, Expert on usability)
- Sebastian Koralewski (Engineer/Tester, Expert on MMRP)