

artdaq\_mfextensions  
1.09.00

Generated by Doxygen 1.8.5

Thu Sep 5 2024 10:58:43



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	mfplugins::ELANSI::Config Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Data Documentation	7
4.1.2.1	bellOnError	7
4.1.2.2	blinkOnError	8
4.1.2.3	debugColor	8
4.1.2.4	errorColor	8
4.1.2.5	infoColor	8
4.1.2.6	warningColor	9
4.2	mfplugins::ELFriendly::Config Struct Reference	9
4.2.1	Detailed Description	9
4.2.2	Member Data Documentation	9
4.2.2.1	delimiter	9
4.3	mfplugins::ELMultiFileOutput::Config Struct Reference	10
4.3.1	Detailed Description	10
4.3.2	Member Data Documentation	10
4.3.2.1	append	10
4.3.2.2	baseDir	10
4.3.2.3	useApplication	11

4.3.2.4	useCategory	11
4.3.2.5	useHostname	11
4.3.2.6	useModule	11
4.4	mfplugins::ELOTS::Config Struct Reference	11
4.4.1	Detailed Description	12
4.4.2	Member Data Documentation	12
4.4.2.1	filename_delimit	12
4.4.2.2	format_string	12
4.5	mfplugins::ELSMTP::Config Struct Reference	12
4.5.1	Detailed Description	13
4.5.2	Member Data Documentation	13
4.5.2.1	from	13
4.5.2.2	host	14
4.5.2.3	messageHeader	14
4.5.2.4	pw	14
4.5.2.5	sendInterval	14
4.5.2.6	subject	14
4.5.2.7	to	15
4.5.2.8	user	15
4.5.2.9	useSmtps	15
4.5.2.10	verifyCert	15
4.6	mfplugins::ELTRACE::Config Struct Reference	15
4.6.1	Detailed Description	16
4.6.2	Member Data Documentation	16
4.6.2.1	lvlm	16
4.6.2.2	lvls	16
4.7	mfplugins::ELUDP::Config Struct Reference	16
4.7.1	Detailed Description	17
4.7.2	Member Data Documentation	17
4.7.2.1	error_max	17
4.7.2.2	error_report	17
4.7.2.3	filename_delimit	18
4.7.2.4	host	18
4.7.2.5	multicast_enabled	18
4.7.2.6	output_address	18
4.8	mfplugins::ELANSI Class Reference	19
4.8.1	Detailed Description	19

4.8.2	Constructor & Destructor Documentation	19
4.8.2.1	ELANSI	19
4.8.3	Member Function Documentation	20
4.8.3.1	routePayload	20
4.9	mfplugins::ELFriendly Class Reference	21
4.9.1	Detailed Description	21
4.9.2	Constructor & Destructor Documentation	22
4.9.2.1	ELFriendly	22
4.9.3	Member Function Documentation	22
4.9.3.1	fillPrefix	22
4.9.3.2	fillSuffix	22
4.9.3.3	fillUsrMsg	22
4.10	mfplugins::ELMultiFileOutput Class Reference	22
4.10.1	Detailed Description	23
4.10.2	Constructor & Destructor Documentation	23
4.10.2.1	ELMultiFileOutput	23
4.10.2.2	~ELMultiFileOutput	24
4.10.3	Member Function Documentation	24
4.10.3.1	routePayload	24
4.11	mfplugins::ELOTS Class Reference	24
4.11.1	Detailed Description	25
4.11.2	Constructor & Destructor Documentation	25
4.11.2.1	ELOTS	25
4.11.3	Member Function Documentation	25
4.11.3.1	fillPrefix	25
4.11.3.2	fillUsrMsg	25
4.11.3.3	routePayload	25
4.12	mfplugins::ELSMTP Class Reference	26
4.12.1	Detailed Description	26
4.12.2	Constructor & Destructor Documentation	27
4.12.2.1	ELSMTP	27
4.12.3	Member Function Documentation	28
4.12.3.1	routePayload	28
4.13	mfplugins::ELTRACE Class Reference	28
4.13.1	Detailed Description	29
4.13.2	Constructor & Destructor Documentation	29
4.13.2.1	ELTRACE	29

4.13.3	Member Function Documentation	29
4.13.3.1	fillPrefix	29
4.13.3.2	fillUsrMsg	29
4.13.3.3	routePayload	30
4.14	mfplugins::ELUDP Class Reference	31
4.14.1	Detailed Description	32
4.14.2	Constructor & Destructor Documentation	32
4.14.2.1	ELUDP	32
4.14.3	Member Function Documentation	32
4.14.3.1	fillPrefix	32
4.14.3.2	fillUsrMsg	32
4.14.3.3	routePayload	32
4.15	mfviewer::LogReader Class Reference	33
4.15.1	Detailed Description	33
4.15.2	Constructor & Destructor Documentation	33
4.15.2.1	LogReader	33
4.15.2.2	~LogReader	34
4.15.3	Member Function Documentation	34
4.15.3.1	iseof	34
4.15.3.2	read_next	34
4.15.3.3	run	34
4.16	msgViewerDlg Class Reference	34
4.16.1	Detailed Description	35
4.16.2	Constructor & Destructor Documentation	35
4.16.2.1	msgViewerDlg	35
4.16.3	Member Function Documentation	36
4.16.3.1	closeEvent	36
4.17	mfviewer::MVReceiver Class Reference	36
4.17.1	Detailed Description	37
4.17.2	Constructor & Destructor Documentation	37
4.17.2.1	MVReceiver	37
4.17.2.2	~MVReceiver	37
4.17.3	Member Function Documentation	38
4.17.3.1	NewMessage	38
4.17.3.2	stop	39
4.17.4	Member Data Documentation	39
4.17.4.1	stopRequested_	39

4.18 qt_mf_msg Class Reference . . . . .	39
4.18.1 Detailed Description . . . . .	40
4.18.2 Constructor & Destructor Documentation . . . . .	40
4.18.2.1 qt_mf_msg . . . . .	40
4.18.3 Member Function Documentation . . . . .	41
4.18.3.1 app . . . . .	41
4.18.3.2 cat . . . . .	41
4.18.3.3 color . . . . .	41
4.18.3.4 host . . . . .	41
4.18.3.5 seq . . . . .	42
4.18.3.6 setEventID . . . . .	42
4.18.3.7 setFileName . . . . .	42
4.18.3.8 setHostAddr . . . . .	42
4.18.3.9 setLineNumber . . . . .	42
4.18.3.10 setMessage . . . . .	42
4.18.3.11 setModule . . . . .	43
4.18.3.12 setSeverity . . . . .	43
4.18.3.13 setSeverityLevel . . . . .	43
4.18.3.14 sev . . . . .	43
4.18.3.15 text . . . . .	43
4.18.3.16 time . . . . .	44
4.18.3.17 updateText . . . . .	44
4.19 mfviewer::ReceiverManager Class Reference . . . . .	44
4.19.1 Detailed Description . . . . .	45
4.19.2 Constructor & Destructor Documentation . . . . .	45
4.19.2.1 ReceiverManager . . . . .	45
4.19.2.2 ~ReceiverManager . . . . .	45
4.19.3 Member Function Documentation . . . . .	45
4.19.3.1 newMessage . . . . .	45
4.19.3.2 start . . . . .	45
4.19.3.3 stop . . . . .	46
4.20 suppress Class Reference . . . . .	46
4.20.1 Detailed Description . . . . .	46
4.20.2 Constructor & Destructor Documentation . . . . .	46
4.20.2.1 suppress . . . . .	46
4.20.3 Member Function Documentation . . . . .	46
4.20.3.1 match . . . . .	46

4.20.3.2	use	47
4.21	throttle Class Reference	47
4.21.1	Detailed Description	47
4.21.2	Constructor & Destructor Documentation	47
4.21.2.1	throttle	47
4.21.3	Member Function Documentation	48
4.21.3.1	reach_limit	48
4.21.3.2	use	48
4.22	mfviewer::UDPReceiver Class Reference	48
4.22.1	Detailed Description	49
4.22.2	Constructor & Destructor Documentation	49
4.22.2.1	UDPReceiver	49
4.22.2.2	~UDPReceiver	49
4.22.3	Member Function Documentation	49
4.22.3.1	read_msg	49
4.22.3.2	run	50
4.22.3.3	validate_packet	50
4.23	upload_status Struct Reference	50
4.23.1	Detailed Description	51
<b>5</b>	<b>File Documentation</b>	<b>53</b>
5.1	artdaq_mfextensions/mfextensions/Destinations/detail/curl_send_message.h File Reference	53
5.1.1	Detailed Description	53
5.1.2	Function Documentation	53
5.1.2.1	send_message	54
5.1.2.2	send_message_ssl	55
5.2	artdaq_mfextensions/mfextensions/Receivers/detail/TCP_listen_fd.hh File Reference	55
5.2.1	Detailed Description	55
5.2.2	Function Documentation	56
5.2.2.1	TCP_listen_fd	56
5.3	artdaq_mfextensions/mfextensions/Receivers/detail/TCPConnect.hh File Reference	56
5.3.1	Detailed Description	57
5.3.2	Function Documentation	57
5.3.2.1	GetInterfaceForNetwork	57
5.3.2.2	ResolveHost	57
5.3.2.3	ResolveHost	57
5.3.2.4	TCPConnect	58



CONTENTS	ix
<a href="#">Index</a>	60



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

mfplugins::ELANSI::Config . . . . .	7
mfplugins::ELFriendly::Config . . . . .	9
mfplugins::ELMultiFileOutput::Config . . . . .	10
mfplugins::ELOTS::Config . . . . .	11
mfplugins::ELSMTP::Config . . . . .	12
mfplugins::ELTRACE::Config . . . . .	15
mfplugins::ELUDP::Config . . . . .	16
ELostreamOutput	
mfplugins::ELFriendly . . . . .	21
MsgViewerDlg	
msgViewerDlg . . . . .	34
QDialog	
msgViewerDlg . . . . .	34
QObject	
mfviewer::ReceiverManager . . . . .	44
qt_mf_msg . . . . .	39
QThread	
mfviewer::MVReceiver . . . . .	36
mfviewer::LogReader . . . . .	33
mfviewer::UDPReceiver . . . . .	48
suppress . . . . .	46
throttle . . . . .	47
upload_status . . . . .	50
ELdestination	
mfplugins::ELANSI . . . . .	19
mfplugins::ELMultiFileOutput . . . . .	22
mfplugins::ELOTS . . . . .	24
mfplugins::ELSMTP . . . . .	26
mfplugins::ELTRACE . . . . .	28
mfplugins::ELUDP . . . . .	31



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">mfplugins::ELANSI::Config</a>	Configuration parameters for <a href="#">ELANSI</a> . . . . .	7
<a href="#">mfplugins::ELFriendly::Config</a>	Configuration Parameters for <a href="#">ELFriendly</a> . . . . .	9
<a href="#">mfplugins::ELMultiFileOutput::Config</a>	Configuration parameters for <a href="#">ELMultiFileOutput</a> . . . . .	10
<a href="#">mfplugins::ELOTS::Config</a>	Configuration Parameters for <a href="#">ELOTS</a> . . . . .	11
<a href="#">mfplugins::ELSMTP::Config</a>	Configuration parameters for <a href="#">ELSMTP</a> . . . . .	12
<a href="#">mfplugins::ELTRACE::Config</a>	Configuration Parameters for <a href="#">ELTRACE</a> . . . . .	15
<a href="#">mfplugins::ELUDP::Config</a>	Configuration Parameters for <a href="#">ELUDP</a> . . . . .	16
<a href="#">mfplugins::ELANSI</a>	Message Facility destination which colorizes the console output . . . . .	19
<a href="#">mfplugins::ELFriendly</a>	Parser-Friendly Message Facility destination plugin . . . . .	21
<a href="#">mfplugins::ELMultiFileOutput</a>	Message Facility Destination which automatically opens files and sorts messages into them based on given criteria . . . . .	22
<a href="#">mfplugins::ELOTS</a>	Message Facility OTS Console Destination Formats messages into Ryan's favorite format for OTS . . . . .	24
<a href="#">mfplugins::ELSMTP</a>	SMTP Message Facility destination plugin (Using libcurl) . . . . .	26
<a href="#">mfplugins::ELTRACE</a>	Message Facility destination which logs messages to a TRACE buffer . . . . .	28
<a href="#">mfplugins::ELUDP</a>	Message Facility UDP Streamer Destination Formats messages into a delimited string and sends via UDP . . . . .	31
<a href="#">mfviewer::LogReader</a>	MessageFacility Log Reader Read messagefacility log archive and reemit as messagefacility messages . . . . .	33

<a href="#">msgViewerDlg</a>	
Message Viewer Dialog Window . . . . .	34
<a href="#">mfviewer::MVReceiver</a>	
A <a href="#">MVReceiver</a> class listens for messages and raises a signal when one arrives . . . . .	36
<a href="#">qt_mf_msg</a>	
Qt wrapper around MessageFacility message . . . . .	39
<a href="#">mfviewer::ReceiverManager</a>	
The <a href="#">ReceiverManager</a> loads one or more receiver plugins and displays messages received by those plugins on the Message Viewer dialog . . . . .	44
<a href="#">suppress</a>	
Suppress messages based on a regular expression . . . . .	46
<a href="#">throttle</a>	
Throttle messages based on name and time limits. Separate from MessageFacility limiting. . . . .	47
<a href="#">mfviewer::UDPReceiver</a>	
Receive messages through a UDP socket. Expects the syslog format provided by UDP_mfPlugin (ELUDP) . . . . .	48
<a href="#">upload_status</a>	
Structure to track progress of upload in cURL send function . . . . .	50

## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

artdaq_mfextensions/mfextensions/Binaries/ <b>MFTest.cc</b>	??
artdaq_mfextensions/mfextensions/Binaries/ <b>msgsender.cc</b>	??
artdaq_mfextensions/mfextensions/Binaries/ <b>msgserver.cc</b>	??
artdaq_mfextensions/mfextensions/Binaries/ <b>msgviewer.cc</b>	??
artdaq_mfextensions/mfextensions/Binaries/ <b>mvdg.cc</b>	??
artdaq_mfextensions/mfextensions/Binaries/ <b>mvdg.hh</b>	??
artdaq_mfextensions/mfextensions/Destinations/ <b>ANSI_mfPlugin.cc</b>	??
artdaq_mfextensions/mfextensions/Destinations/ <b>Friendly_mfPlugin.cc</b>	??
artdaq_mfextensions/mfextensions/Destinations/ <b>MultiFile_mfPlugin.cc</b>	??
artdaq_mfextensions/mfextensions/Destinations/ <b>OTS_mfPlugin.cc</b>	??
artdaq_mfextensions/mfextensions/Destinations/ <b>SMTP_mfPlugin.cc</b>	??
artdaq_mfextensions/mfextensions/Destinations/ <b>TRACE_mfPlugin.cc</b>	??
artdaq_mfextensions/mfextensions/Destinations/ <b>UDP_mfPlugin.cc</b>	??
artdaq_mfextensions/mfextensions/Destinations/detail/ <b>curl_send_message.c</b>	??
artdaq_mfextensions/mfextensions/Destinations/detail/ <b>curl_send_message.h</b>	53
artdaq_mfextensions/mfextensions/Extensions/ <b>suppress.cc</b>	??
artdaq_mfextensions/mfextensions/Extensions/ <b>suppress.hh</b>	??
artdaq_mfextensions/mfextensions/Extensions/ <b>throttle.cc</b>	??
artdaq_mfextensions/mfextensions/Extensions/ <b>throttle.hh</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>LogReader_receiver.cc</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>LogReader_receiver.hh</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>makeMVReceiver.cc</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>makeMVReceiver.hh</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>MVReceiver.cc</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>MVReceiver.hh</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>qt_mf_msg.cc</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>qt_mf_msg.hh</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>ReceiverMacros.hh</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>ReceiverManager.cc</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>ReceiverManager.hh</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>UDP_receiver.cc</b>	??
artdaq_mfextensions/mfextensions/Receivers/ <b>UDP_receiver.hh</b>	??
artdaq_mfextensions/mfextensions/Receivers/detail/ <b>TCP_listen_fd.hh</b>	55
artdaq_mfextensions/mfextensions/Receivers/detail/ <b>TCPConnect.hh</b>	56

artdaq_mfextensions/test/Extensions/ <b>suppress_t.cc</b>	??
artdaq_mfextensions/test/Extensions/ <b>throttle_t.cc</b>	??
artdaq_mfextensions/tools/ <b>mf_simple.cc</b>	??
artdaq_mfextensions/tools/ <b>udp_send_mfmsg.py</b>	??



## Chapter 4

# Class Documentation

### 4.1 mfplugins::ELANSI::Config Struct Reference

Configuration parameters for [ELANSI](#).

#### Public Attributes

- fhicl::TableFragment  
< ELdestination::Config > [elDestConfig](#)  
*ELdestination common config parameters.*
- fhicl::Atom< bool > [bellOnError](#)  
*"bell\_on\_error" (Default: true): Whether to ring the system bell on error messages*
- fhicl::Atom< bool > [blinkOnError](#)  
*"blink\_error\_messages" (Default: false): Whether to print error messages with blinking text*
- fhicl::Atom< std::string > [errorColor](#)  
*"error\_ansi\_color" (Default: "\033[1m\033[91m"): ANSI Color string for Error Messages*
- fhicl::Atom< std::string > [warningColor](#)  
*"warning\_ansi\_color" (Default: "\033[1m\033[93m"): ANSI Color string for Warning Messages*
- fhicl::Atom< std::string > [infoColor](#)  
*"info\_ansi\_color" (Default: "\033[92m"): ANSI Color string for Info Messages*
- fhicl::Atom< std::string > [debugColor](#)  
*"debug\_ansi\_color" (Default: "\033[39m"): ANSI Color string for ErrDebugger Messages*

#### 4.1.1 Detailed Description

Configuration parameters for [ELANSI](#).

Definition at line 26 of file ANSI\_mfPlugin.cc.

#### 4.1.2 Member Data Documentation

##### 4.1.2.1 fhicl::Atom<bool> mfplugins::ELANSI::Config::bellOnError

**Initial value:**

```
= fhicl::Atom<bool>{
    fhicl::Name{"bell_on_error"}, fhicl::Comment{"Whether to ring the system bell on error messages"}, true}
```

"bell\_on\_error" (Default: true): Whether to ring the system bell on error messages

Definition at line 31 of file ANSI\_mfPlugin.cc.

#### 4.1.2.2 fhicl::Atom<bool> mfplugins::ELANSI::Config::blinkOnError

##### Initial value:

```
=
    fhicl::Atom<bool>{fhicl::Name{"blink_error_messages"},
        fhicl::Comment{"Whether to print error messages with blinking text"}, false}
```

"blink\_error\_messages" (Default: false): Whether to print error messages with blinking text

Definition at line 34 of file ANSI\_mfPlugin.cc.

#### 4.1.2.3 fhicl::Atom<std::string> mfplugins::ELANSI::Config::debugColor

##### Initial value:

```
= fhicl::Atom<std::string>{
    fhicl::Name{"debug_ansi_color"}, fhicl::Comment{"ANSI Color string for Debug Messages"}, "\033[39m"}
```

"debug\_ansi\_color" (Default: "\033[39m"): ANSI Color string for ErrDebugger Messages

Definition at line 47 of file ANSI\_mfPlugin.cc.

#### 4.1.2.4 fhicl::Atom<std::string> mfplugins::ELANSI::Config::errorColor

##### Initial value:

```
= fhicl::Atom<std::string>{
    fhicl::Name{"error_ansi_color"}, fhicl::Comment{"ANSI Color string for Error Messages"}, "\033[1m\033[91m"}
```

"error\_ansi\_color" (Default: "\033[1m\033[91m"): ANSI Color string for Error Messages

Definition at line 38 of file ANSI\_mfPlugin.cc.

#### 4.1.2.5 fhicl::Atom<std::string> mfplugins::ELANSI::Config::infoColor

##### Initial value:

```
= fhicl::Atom<std::string>{
    fhicl::Name{"info_ansi_color"}, fhicl::Comment{"ANSI Color string for Info Messages"}, "\033[92m"}
```

"info\_ansi\_color" (Default: "\033[92m"): ANSI Color string for Info Messages

Definition at line 44 of file ANSI\_mfPlugin.cc.

## 4.1.2.6 fhicl::Atom&lt;std::string&gt; mfplugins::ELANSI::Config::warningColor

**Initial value:**

```
= fhicl::Atom<std::string>{
    fhicl::Name{"warning_ansi_color"}, fhicl::Comment{"ANSI Color string for Warning Messages"}, "
    \033[1m\033[93m"}
```

"warning\_ansi\_color" (Default: "\033[1m\033[93m"): ANSI Color string for Warning Messages

Definition at line 41 of file ANSI\_mfPlugin.cc.

The documentation for this struct was generated from the following file:

- artdaq\_mfextensions/mfextensions/Destinations/ANSI\_mfPlugin.cc

## 4.2 mfplugins::ELFriendly::Config Struct Reference

Configuration Parameters for [ELFriendly](#).

**Public Attributes**

- fhicl::TableFragment  
< ELOstreamOutput::Config > [eLOstrConfig](#)  
*Configuration parameters for ELOstreamOutput.*
- fhicl::Atom< std::string > [delimiter](#)  
*"field\_delimiter" (Default: " "): String to print between each message field*

### 4.2.1 Detailed Description

Configuration Parameters for [ELFriendly](#).

Definition at line 30 of file Friendly\_mfPlugin.cc.

### 4.2.2 Member Data Documentation

## 4.2.2.1 fhicl::Atom&lt;std::string&gt; mfplugins::ELFriendly::Config::delimiter

**Initial value:**

```
= fhicl::Atom<std::string>{
    fhicl::Name{"field_delimiter"}, fhicl::Comment{"String to print between each message field"}, "
    "}
```

"field\_delimiter" (Default: " "): String to print between each message field

Definition at line 35 of file Friendly\_mfPlugin.cc.

The documentation for this struct was generated from the following file:

- artdaq\_mfextensions/mfextensions/Destinations/Friendly\_mfPlugin.cc

## 4.3 mfplugins::ELMultiFileOutput::Config Struct Reference

Configuration parameters for [ELMultiFileOutput](#).

### Public Attributes

- fhicl::TableFragment  
< ELdestination::Config > [elDestConfig](#)  
*ELdestination common config parameters.*
- fhicl::Atom< std::string > [baseDir](#)  
*"base\_directory" (Default: "/tmp"): Directory where log files will be created*
- fhicl::Atom< bool > [append](#)  
*"append" (Default: true): Append to existing log files*
- fhicl::Atom< bool > [useHostname](#)  
*"use\_hostname" (Default: true): Use the hostname when generating log file names*
- fhicl::Atom< bool > [useApplication](#)  
*"use\_application" (Default: true): Use the application field when generating log file names*
- fhicl::Atom< bool > [useCategory](#)  
*"use\_category" (Default: false): Use the category field when generating log file names*
- fhicl::Atom< bool > [useModule](#)  
*"use\_module" (Default: false): Use the module field when generating log file names*

### 4.3.1 Detailed Description

Configuration parameters for [ELMultiFileOutput](#).

Definition at line 26 of file MultiFile\_mfPlugin.cc.

### 4.3.2 Member Data Documentation

#### 4.3.2.1 fhicl::Atom<bool> mfplugins::ELMultiFileOutput::Config::append

**Initial value:**

```
=
    fhicl::Atom<bool>{fhicl::Name{"append"}, fhicl::Comment{"Append to existing log files"}, true}
```

"append" (Default: true): Append to existing log files

Definition at line 34 of file MultiFile\_mfPlugin.cc.

#### 4.3.2.2 fhicl::Atom<std::string> mfplugins::ELMultiFileOutput::Config::baseDir

**Initial value:**

```
= fhicl::Atom<std::string>{
    fhicl::Name{"base_directory"}, fhicl::Comment{"Directory where log files will be created"}, "
    /tmp" }
```

"base\_directory" (Default: "/tmp"): Directory where log files will be created

Definition at line 31 of file MultiFile\_mfPlugin.cc.

## 4.3.2.3 fhicl::Atom&lt;bool&gt; mfplugins::ELMultiFileOutput::Config::useApplication

**Initial value:**

```
=
    fhicl::Atom<bool>{fhicl::Name{"use_application"},
                      fhicl::Comment{"Use the application field when generating log file names"},
                      true}
```

"use\_application" (Default: true): Use the application field when generating log file names

Definition at line 40 of file MultiFile\_mfPlugin.cc.

## 4.3.2.4 fhicl::Atom&lt;bool&gt; mfplugins::ELMultiFileOutput::Config::useCategory

**Initial value:**

```
= fhicl::Atom<bool>{
    fhicl::Name{"use_category"}, fhicl::Comment{"Use the category field when generating log file
names"}, false}
```

"use\_category" (Default: false): Use the category field when generating log file names

Definition at line 44 of file MultiFile\_mfPlugin.cc.

## 4.3.2.5 fhicl::Atom&lt;bool&gt; mfplugins::ELMultiFileOutput::Config::useHostname

**Initial value:**

```
= fhicl::Atom<bool>{
    fhicl::Name{"use_hostname"}, fhicl::Comment{"Use the hostname when generating log file names"},
    true}
```

"use\_hostname" (Default: true): Use the hostname when generating log file names

Definition at line 37 of file MultiFile\_mfPlugin.cc.

## 4.3.2.6 fhicl::Atom&lt;bool&gt; mfplugins::ELMultiFileOutput::Config::useModule

**Initial value:**

```
= fhicl::Atom<bool>{
    fhicl::Name{"use_module"}, fhicl::Comment{"Use the module field when generating log file names"},
    false}
```

"use\_module" (Default: false): Use the module field when generating log file names

Definition at line 47 of file MultiFile\_mfPlugin.cc.

The documentation for this struct was generated from the following file:

- artdaq\_mfextensions/mfextensions/Destinations/MultiFile\_mfPlugin.cc

## 4.4 mfplugins::ELOTS::Config Struct Reference

Configuration Parameters for [ELOTS](#).

## Public Attributes

- fhicl::TableFragment  
 < ELdestination::Config > [elDestConfig](#)  
*ELDestination common config parameters.*
- fhicl::Atom< std::string > [format\\_string](#)  
*format\_string (Default: "%L:%N:%f [%u] %m"): Format specifier for printing to console. %% => " ...*
- fhicl::Atom< std::string > [filename\\_delimit](#)  
*filename\_delimit (Default: "/"): Grab path after this. "/srcs/" /x/srcs/y/z.cc => y/z.cc*

### 4.4.1 Detailed Description

Configuration Parameters for [ELOTS](#).

Definition at line 42 of file OTS\_mfPlugin.cc.

### 4.4.2 Member Data Documentation

#### 4.4.2.1 fhicl::Atom<std::string> mfplugins::ELOTS::Config::filename\_delimit

**Initial value:**

```
=
    fhicl::Atom<std::string>{fhicl::Name{"filename_delimit"},
                             fhicl::Comment{"Grab path after this. \"/srcs/" /x/srcs/y/z.cc =>
y/z.cc"}, "/"}
```

filename\_delimit (Default: "/"): Grab path after this. "/srcs/" /x/srcs/y/z.cc => y/z.cc

Definition at line 51 of file OTS\_mfPlugin.cc.

#### 4.4.2.2 fhicl::Atom<std::string> mfplugins::ELOTS::Config::format\_string

**Initial value:**

```
= fhicl::Atom<std::string>{
    fhicl::Name{"format_string"}, fhicl::Comment{"Format specifier for printing to console. %% =>
'%' ... "},
    "%L:%N:%f [%u] %m"}
```

format\_string (Default: "%L:%N:%f [%u] %m"): Format specifier for printing to console. %% => " ...

Definition at line 47 of file OTS\_mfPlugin.cc.

The documentation for this struct was generated from the following file:

- artdaq\_mfextensions/mfextensions/Destinations/OTS\_mfPlugin.cc

## 4.5 mfplugins::ELSMTP::Config Struct Reference

Configuration parameters for [ELSMTP](#).

## Public Attributes

- fhicl::TableFragment  
< ELdestination::Config > [elDestConfig](#)  
*ELDestination common config parameters.*
- fhicl::Atom< std::string > [host](#)  
*"host" (Default: "smtp.fnal.gov"): SMTP Server hostname*
- fhicl::Atom< int > [port](#) = fhicl::Atom<int>{fhicl::Name{"port"}, fhicl::Comment{"SMTP Server port"}, 25}  
*"port" (Default: 25): SMTP Server port*
- fhicl::Sequence< std::string > [to](#)  
*"to\_addresses" (Default: {}): The list of email addresses that SMTP mfPlugin should sent to*
- fhicl::Atom< std::string > [from](#)  
*"from\_address" (REQUIRED): Source email address*
- fhicl::Atom< std::string > [subject](#)  
*"subject" (Default: "MessageFacility SMTP Message Digest"): Subject of the email message*
- fhicl::Atom< std::string > [messageHeader](#)  
*"message\_header" (Default: ""): String to preface messages with in email body*
- fhicl::Atom< bool > [useSmtps](#)  
*"use\_smtps" (Default: false): Use SMTPS protocol*
- fhicl::Atom< std::string > [user](#)  
*"smtp\_username" (Default: ""): Username for SMTP server*
- fhicl::Atom< std::string > [pw](#)  
*"smtp\_password" (Default: ""): Password for SMTP server*
- fhicl::Atom< bool > [verifyCert](#)  
*"verify\_host\_ssl\_certificate" (Default: true): Whether to run full SSL verify on SMTP server in SMTPS mode*
- fhicl::Atom< size\_t > [sendInterval](#)  
*"email\_send\_interval\_seconds" (Default: 15): Only send email every N seconds*

### 4.5.1 Detailed Description

Configuration parameters for [ELSMTP](#).

Definition at line 42 of file SMTP\_mfPlugin.cc.

### 4.5.2 Member Data Documentation

#### 4.5.2.1 fhicl::Atom<std::string> mfplugins::ELSMTP::Config::from

**Initial value:**

```
=
    fhicl::Atom<std::string>{fhicl::Name{"from_address"}, fhicl::Comment{"Source email address"}}
```

"from\_address" (REQUIRED): Source email address

Definition at line 56 of file SMTP\_mfPlugin.cc.

#### 4.5.2.2 fhicl::Atom<std::string> mfplugins::ELSMTP::Config::host

##### Initial value:

```
=
    fhicl::Atom<std::string>{fhicl::Name{"host"}, fhicl::Comment{"SMTP Server hostname"}, "
    smtp.fnal.gov"}
```

"host" (Default: "smtp.fnal.gov"): SMTP Server hostname

Definition at line 47 of file SMTP\_mfPlugin.cc.

#### 4.5.2.3 fhicl::Atom<std::string> mfplugins::ELSMTP::Config::messageHeader

##### Initial value:

```
= fhicl::Atom<std::string>{
    fhicl::Name{"message_header"}, fhicl::Comment{"String to preface messages with in email body"},
    ""}
```

"message\_header" (Default: ""): String to preface messages with in email body

Definition at line 62 of file SMTP\_mfPlugin.cc.

#### 4.5.2.4 fhicl::Atom<std::string> mfplugins::ELSMTP::Config::pw

##### Initial value:

```
=
    fhicl::Atom<std::string>{fhicl::Name{"smtp_password"}, fhicl::Comment{"Password for SMTP server
    "}, ""}
```

"smtp\_password" (Default: ""): Password for SMTP server

Definition at line 71 of file SMTP\_mfPlugin.cc.

#### 4.5.2.5 fhicl::Atom<size\_t> mfplugins::ELSMTP::Config::sendInterval

##### Initial value:

```
= fhicl::Atom<size_t>{fhicl::Name{"email_send_interval_seconds"},
    fhicl::Comment{"Only send email every N
    seconds"}, 15}
```

"email\_send\_interval\_seconds" (Default: 15): Only send email every N seconds

Definition at line 78 of file SMTP\_mfPlugin.cc.

#### 4.5.2.6 fhicl::Atom<std::string> mfplugins::ELSMTP::Config::subject

##### Initial value:

```
= fhicl::Atom<std::string>{
    fhicl::Name{"subject"}, fhicl::Comment{"Subject of the email message"}, "MessageFacility SMTP
    Message Digest"}
```

"subject" (Default: "MessageFacility SMTP Message Digest"): Subject of the email message

Definition at line 59 of file SMTP\_mfPlugin.cc.



## 4.5.2.7 fhicl::Sequence&lt;std::string&gt; mfplugins::ELSMTP::Config::to

**Initial value:**

```
= fhicl::Sequence<std::string>{
    fhicl::Name{"to_addresses"}, fhicl::Comment{"The list of email addresses that SMTP mfPlugin
    should sent to"},
    fhicl::Sequence<std::string>::default_type{}}
```

"to\_addresses" (Default: {}): The list of email addresses that SMTP mfPlugin should sent to

Definition at line 52 of file SMTP\_mfPlugin.cc.

## 4.5.2.8 fhicl::Atom&lt;std::string&gt; mfplugins::ELSMTP::Config::user

**Initial value:**

```
=
    fhicl::Atom<std::string>{fhicl::Name{"smtp_username"}, fhicl::Comment{"Username for SMTP server
    "}, ""}
```

"smtp\_username" (Default: ""): Username for SMTP server

Definition at line 68 of file SMTP\_mfPlugin.cc.

## 4.5.2.9 fhicl::Atom&lt;bool&gt; mfplugins::ELSMTP::Config::useSmtps

**Initial value:**

```
=
    fhicl::Atom<bool>{fhicl::Name{"use_smtps"}, fhicl::Comment{"Use SMTPS protocol"}, false}
```

"use\_smtps" (Default: false): Use SMTPS protocol

Definition at line 65 of file SMTP\_mfPlugin.cc.

## 4.5.2.10 fhicl::Atom&lt;bool&gt; mfplugins::ELSMTP::Config::verifyCert

**Initial value:**

```
=
    fhicl::Atom<bool>{fhicl::Name{"verify_host_ssl_certificate"},
    fhicl::Comment{"Whether to run full SSL verify on SMTP server in SMTPS mode"},
    true}
```

"verify\_host\_ssl\_certificate" (Default: true): Whether to run full SSL verify on SMTP server in SMTPS mode

Definition at line 74 of file SMTP\_mfPlugin.cc.

The documentation for this struct was generated from the following file:

- artdaq\_mfextensions/mfextensions/Destinations/SMTP\_mfPlugin.cc

## 4.6 mfplugins::ELTRACE::Config Struct Reference

Configuration Parameters for [ELTRACE](#).

## Public Attributes

- fhicl::TableFragment  
< ELdestination::Config > [elDestConfig](#)  
*ELDestination common parameters.*
- fhicl::Atom< size\_t > [lvls](#)  
*"lvls" (Default: 0): TRACE level mask for Slow output*
- fhicl::Atom< size\_t > [lvlm](#)  
*"lvlm" (Default: 0): TRACE level mask for Memory output*

### 4.6.1 Detailed Description

Configuration Parameters for [ELTRACE](#).

Definition at line 38 of file TRACE\_mfPlugin.cc.

### 4.6.2 Member Data Documentation

#### 4.6.2.1 fhicl::Atom<size\_t> mfplugins::ELTRACE::Config::lvlm

##### Initial value:

```
=
    fhicl::Atom<size_t>{fhicl::Name{"lvlm"}, fhicl::Comment{"TRACE level mask for Memory output"},
    0}
```

"lvlm" (Default: 0): TRACE level mask for Memory output

Definition at line 46 of file TRACE\_mfPlugin.cc.

#### 4.6.2.2 fhicl::Atom<size\_t> mfplugins::ELTRACE::Config::lvls

##### Initial value:

```
=
    fhicl::Atom<size_t>{fhicl::Name{"lvls"}, fhicl::Comment{"TRACE level mask for Slow output"}, 0}
```

"lvls" (Default: 0): TRACE level mask for Slow output

Definition at line 43 of file TRACE\_mfPlugin.cc.

The documentation for this struct was generated from the following file:

- artdaq\_mfextensions/mfextensions/Destinations/TRACE\_mfPlugin.cc

## 4.7 mfplugins::ELUDP::Config Struct Reference

Configuration Parameters for [ELUDP](#).

## Public Attributes

- fhicl::TableFragment  
< ELdestination::Config > [elDestConfig](#)  
*ELDestination common config parameters.*
- fhicl::Atom< int > [error\\_max](#)
- fhicl::Atom< int > [error\\_report](#)  
*"error\_report\_backoff\_factor" (Default: 100): Print an error message every N errors*
- fhicl::Atom< std::string > [host](#)  
*"host" (Default: "227.128.12.27"): Address to send messages to*
- fhicl::Atom< int > [port](#) = fhicl::Atom<int>{fhicl::Name{"port"}, fhicl::Comment{"Port to send messages to"}, 5140}  
*"port" (Default: 5140): Port to send messages to*
- fhicl::Atom< bool > [multicast\\_enabled](#)  
*"multicast\_enabled" (Default: false): Whether messages should be sent via multicast*
- fhicl::Atom< std::string > [output\\_address](#)
- fhicl::Atom< std::string > [filename\\_delimit](#)  
*filename\_delimit (Default: "/"): Grab path after this. "/srcs/" /x/srcs/y/z.cc => y/z.cc*

### 4.7.1 Detailed Description

Configuration Parameters for [ELUDP](#).

Definition at line 43 of file UDP\_mfPlugin.cc.

### 4.7.2 Member Data Documentation

#### 4.7.2.1 fhicl::Atom<int> mfplugins::ELUDP::Config::error\_max

**Initial value:**

```
= fhicl::Atom<int>{
    fhicl::Name{"error_turnoff_threshold"},
    fhicl::Comment{"Number of errors before turning off destination (default: 0, don't turn off)"},
    0}
```

"error\_turnoff\_threshold" (Default: 0): Number of errors before turning off destination (default: 0, don't turn off)"

Definition at line 49 of file UDP\_mfPlugin.cc.

#### 4.7.2.2 fhicl::Atom<int> mfplugins::ELUDP::Config::error\_report

**Initial value:**

```
= fhicl::Atom<int>{fhicl::Name{"error_report_backoff_factor"},
    fhicl::Comment{"Print an error message every N
    errors"}, 100}
```

"error\_report\_backoff\_factor" (Default: 100): Print an error message every N errors

Definition at line 53 of file UDP\_mfPlugin.cc.

#### 4.7.2.3 fhicl::Atom<std::string> mfplugins::ELUDP::Config::filename\_delimit

##### Initial value:

```
=
    fhicl::Atom<std::string>{fhicl::Name{"filename_delimit"},
                             fhicl::Comment{"Grab path after this. \"/srcs/" /x/srcs/y/z.cc =>
y/z.cc. NOTE: only works if full filename is given to this plugin (based on which mf::<method> is used)."}, "/"
}
```

filename\_delimit (Default: "/"): Grab path after this. "/srcs/" /x/srcs/y/z.cc => y/z.cc

Definition at line 69 of file UDP\_mfPlugin.cc.

#### 4.7.2.4 fhicl::Atom<std::string> mfplugins::ELUDP::Config::host

##### Initial value:

```
=
    fhicl::Atom<std::string>{fhicl::Name{"host"}, fhicl::Comment{"Address to send messages to"}, "
227.128.12.27"}
```

"host" (Default: "227.128.12.27"): Address to send messages to

Definition at line 56 of file UDP\_mfPlugin.cc.

#### 4.7.2.5 fhicl::Atom<bool> mfplugins::ELUDP::Config::multicast\_enabled

##### Initial value:

```
= fhicl::Atom<bool>{
    fhicl::Name{"multicast_enabled"}, fhicl::Comment{"Whether messages should be sent via multicast
"}, false}
```

"multicast\_enabled" (Default: false): Whether messages should be sent via multicast

Definition at line 61 of file UDP\_mfPlugin.cc.

#### 4.7.2.6 fhicl::Atom<std::string> mfplugins::ELUDP::Config::output\_address

##### Initial value:

```
= fhicl::Atom<std::string>{
    fhicl::Name{"multicast_interface_ip"},
    fhicl::Comment{"Use this hostname for multicast output(to assign to the proper NIC)"}, "0.0.0.0
"}
}
```

"multicast\_interface\_ip" (Default: "0.0.0.0"): Use this hostname for multicast output (to assign to the proper NIC)

Definition at line 65 of file UDP\_mfPlugin.cc.

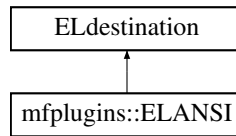
The documentation for this struct was generated from the following file:

- artdaq\_mfextensions/mfextensions/Destinations/UDP\_mfPlugin.cc

## 4.8 mfplugins::ELANSI Class Reference

Message Facility destination which colorizes the console output

Inheritance diagram for mfplugins::ELANSI:



### Classes

- struct [Config](#)  
Configuration parameters for [ELANSI](#).

### Public Types

- using [Parameters](#) = fhicl::WrappedTable< [Config](#) >  
Used for *ParameterSet* validation.

### Public Member Functions

- [ELANSI](#) ([Parameters](#) const &pset)  
[ELANSI](#) Constructor
- void [routePayload](#) (const std::ostream &o, const ErrorObj &msg) override  
Serialize a *MessageFacility* message to the output.

#### 4.8.1 Detailed Description

Message Facility destination which colorizes the console output

Definition at line 20 of file ANSI\_mfPlugin.cc.

#### 4.8.2 Constructor & Destructor Documentation

##### 4.8.2.1 mfplugins::ELANSI::ELANSI ( [Parameters](#) const & *pset* )

[ELANSI](#) Constructor

Parameters

<i>pset</i>	ParameterSet used to configure <a href="#">ELANSI</a>
-------------	---

Definition at line 84 of file ANSI\_mfPlugin.cc.

### 4.8.3 Member Function Documentation

4.8.3.1 `void mfplugins::ELANSI::routePayload ( const std::ostream & o, const ErrorObj & msg )` [override]

Serialize a MessageFacility message to the output.

## Parameters

<i>o</i>	Stringstream object containing message data
<i>msg</i>	MessageFacility object containing header information

Definition at line 93 of file ANSI\_mfPlugin.cc.

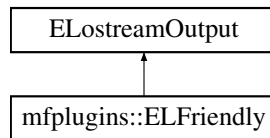
The documentation for this class was generated from the following file:

- artdaq\_mfextensions/mfextensions/Destinations/ANSI\_mfPlugin.cc

## 4.9 mfplugins::ELFriendly Class Reference

Parser-Friendly Message Facility destination plugin

Inheritance diagram for mfplugins::ELFriendly:



### Classes

- struct [Config](#)  
Configuration Parameters for [ELFriendly](#).

### Public Types

- using [Parameters](#) = fhicl::WrappedTable< [Config](#) >  
Used for *ParameterSet* validation.

### Public Member Functions

- [ELFriendly](#) ([Parameters](#) const &pset)  
*ELFriendly* Constructor
- void [fillPrefix](#) (std::ostream &o, const ErrorObj &msg) override  
Fill the "Prefix" portion of the message.
- void [fillUsrMsg](#) (std::ostream &o, const ErrorObj &msg) override  
Fill the "User Message" portion of the message.
- void [fillSuffix](#) (std::ostream &o, const ErrorObj &msg) override  
Fill the "Suffix" portion of the message.

#### 4.9.1 Detailed Description

Parser-Friendly Message Facility destination plugin

Definition at line 24 of file Friendly\_mfPlugin.cc.

## 4.9.2 Constructor & Destructor Documentation

4.9.2.1 `mfplugins::ELFriendly::ELFriendly ( Parameters const & pset ) [explicit]`

[ELFriendly](#) Constructor

Parameters

<i>pset</i>	ParameterSet used to configure <a href="#">ELFriendly</a>
-------------	---

Definition at line 79 of file Friendly\_mfPlugin.cc.

## 4.9.3 Member Function Documentation

4.9.3.1 `void mfplugins::ELFriendly::fillPrefix ( std::ostream & o, const ErrorObj & msg ) [override]`

Fill the "Prefix" portion of the message.

Parameters

<i>o</i>	Output stringstream
<i>msg</i>	MessageFacility object containing header information

Definition at line 85 of file Friendly\_mfPlugin.cc.

4.9.3.2 `void mfplugins::ELFriendly::fillSuffix ( std::ostream & o, const ErrorObj & msg ) [override]`

Fill the "Suffix" portion of the message.

Parameters

<i>o</i>	Output stringstream
<i>msg</i>	MessageFacility object containing header information

Definition at line 235 of file Friendly\_mfPlugin.cc.

4.9.3.3 `void mfplugins::ELFriendly::fillUsrMsg ( std::ostream & o, const ErrorObj & msg ) [override]`

Fill the "User Message" portion of the message.

Parameters

<i>o</i>	Output stringstream
<i>msg</i>	MessageFacility object containing header information

Definition at line 189 of file Friendly\_mfPlugin.cc.

The documentation for this class was generated from the following file:

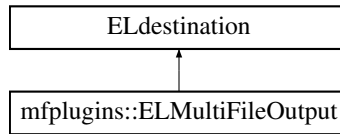
- `artdaq_mfextensions/mfextensions/Destinations/Friendly_mfPlugin.cc`

## 4.10 mfplugins::ELMultiFileOutput Class Reference

Message Facility Destination which automatically opens files and sorts messages into them based on given criteria



Inheritance diagram for mfplugins::ELMultiFileOutput:



## Classes

- struct [Config](#)  
*Configuration parameters for [ELMultiFileOutput](#).*

## Public Types

- using [Parameters](#) = fhicl::WrappedTable< [Config](#) >  
*Used for [ParameterSet](#) validation.*

## Public Member Functions

- [ELMultiFileOutput](#) ([Parameters](#) const &pset)  
*[ELMultiFileOutput](#) Constructor*
- [~ELMultiFileOutput](#) () override=default  
*Default virtual Destructor*
- void [routePayload](#) (const std::ostream &oss, const ErrorObj &msg) override  
*Serialize a MessageFacility message to the output.*
- void [flush](#) () override  
*Flush any text in the ostream buffer to disk.*

### 4.10.1 Detailed Description

Message Facility Destination which automatically opens files and sorts messages into them based on given criteria  
Definition at line 20 of file MultiFile\_mfPlugin.cc.

### 4.10.2 Constructor & Destructor Documentation

4.10.2.1 mfplugins::ELMultiFileOutput::ELMultiFileOutput ( [Parameters](#) const & *pset* ) `[explicit]`

[ELMultiFileOutput](#) Constructor

Parameters

<i>pset</i>	ParameterSet used to configure <a href="#">ELMultiFileOutput</a>
-------------	--

Definition at line 100 of file MultiFile\_mfPlugin.cc.

4.10.2.2 `mfplugins::ELMultiFileOutput::~~ELMultiFileOutput ( ) [override],[default]`

Default virtual Destructor

### 4.10.3 Member Function Documentation

4.10.3.1 `void mfplugins::ELMultiFileOutput::routePayload ( const std::ostringstream & oss, const ErrorObj & msg ) [override]`

Serialize a MessageFacility message to the output.

Parameters

<code>oss</code>	Stringstream object containing message data
<code>msg</code>	MessageFacility object containing header information

Definition at line 106 of file MultiFile\_mfPlugin.cc.

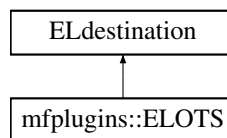
The documentation for this class was generated from the following file:

- `artdaq_mfextensions/mfextensions/Destinations/MultiFile_mfPlugin.cc`

## 4.11 mfplugins::ELOTS Class Reference

Message Facility OTS Console Destination Formats messages into Ryan's favorite format for OTS

Inheritance diagram for `mfplugins::ELOTS`:



### Classes

- struct [Config](#)  
*Configuration Parameters for [ELOTS](#).*

### Public Types

- using [Parameters](#) = `fhicl::WrappedTable< Config >`  
*Used for [ParameterSet](#) validation.*

### Public Member Functions

- [ELOTS](#) ([Parameters](#) const &pset)  
*[ELOTS](#) Constructor*
- void [fillPrefix](#) (std::ostringstream &o, const ErrorObj &msg) *override*

Fill the "Prefix" portion of the message.

- void [fillUsrMsg](#) (std::ostream &o, const ErrorObj &msg) override

Fill the "User Message" portion of the message.

- void [fillSuffix](#) (std::ostream &, const ErrorObj &) override

Fill the "Suffix" portion of the message (Unused)

- void [routePayload](#) (const std::ostream &o, const ErrorObj &e) override

Serialize a MessageFacility message to the output.

### 4.11.1 Detailed Description

Message Facility OTS Console Destination Formats messages into Ryan's favorite format for OTS

Definition at line 36 of file OTS\_mfPlugin.cc.

### 4.11.2 Constructor & Destructor Documentation

#### 4.11.2.1 mfplugins::ELOTS::ELOTS ( Parameters const & pset )

[ELOTS](#) Constructor

Parameters

<i>pset</i>	ParameterSet used to configure <a href="#">ELOTS</a>
-------------	--

Definition at line 109 of file OTS\_mfPlugin.cc.

### 4.11.3 Member Function Documentation

#### 4.11.3.1 void mfplugins::ELOTS::fillPrefix ( std::ostream &o, const ErrorObj &msg ) [override]

Fill the "Prefix" portion of the message.

Parameters

<i>o</i>	Output stringstream
<i>msg</i>	MessageFacility object containing header information

Definition at line 197 of file OTS\_mfPlugin.cc.

#### 4.11.3.2 void mfplugins::ELOTS::fillUsrMsg ( std::ostream &o, const ErrorObj &msg ) [override]

Fill the "User Message" portion of the message.

Parameters

<i>o</i>	Output stringstream
<i>msg</i>	MessageFacility object containing header information

Definition at line 334 of file OTS\_mfPlugin.cc.

#### 4.11.3.3 void mfplugins::ELOTS::routePayload ( const std::ostream &o, const ErrorObj &e ) [override]

Serialize a MessageFacility message to the output.

## Parameters

<i>o</i>	Stringstream object containing message data
<i>e</i>	MessageFacility object containing header information

Definition at line 343 of file OTS\_mfPlugin.cc.

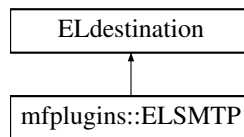
The documentation for this class was generated from the following file:

- `artdaq_mfextensions/mfextensions/Destinations/OTS_mfPlugin.cc`

## 4.12 mfplugins::ELSMTP Class Reference

SMTP Message Facility destination plugin (Using libcurl)

Inheritance diagram for mfplugins::ELSMTP:



## Classes

- struct [Config](#)  
Configuration parameters for [ELSMTP](#).

## Public Types

- using [Parameters](#) = fhicl::WrappedTable< [Config](#) >  
Used for *ParameterSet* validation.

## Public Member Functions

- [ELSMTP](#) ([Parameters](#) const &pset)  
*ELSMTP* Constructor
- virtual void [routePayload](#) (const std::ostream &o, const ErrorObj &msg) override  
Serialize a *MessageFacility* message to the output.

### 4.12.1 Detailed Description

SMTP Message Facility destination plugin (Using libcurl)

Definition at line 36 of file SMTP\_mfPlugin.cc.

## 4.12.2 Constructor & Destructor Documentation

### 4.12.2.1 mfplugins::ELSMTP::ELSMTP ( Parameters const & *pset* )

[ELSMTP](#) Constructor

## Parameters

<i>pset</i>	ParameterSet used to configure <a href="#">ELSMTP</a>
-------------	---

Definition at line 142 of file SMTP\_mfPlugin.cc.

### 4.12.3 Member Function Documentation

4.12.3.1 `void mfplugins::ELSMTP::routePayload ( const std::ostream & o, const ErrorObj & msg )` [override],  
[virtual]

Serialize a MessageFacility message to the output.

## Parameters

<i>o</i>	Stringstream object containing message data
<i>msg</i>	MessageFacility object containing header information

Definition at line 258 of file SMTP\_mfPlugin.cc.

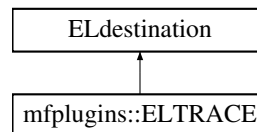
The documentation for this class was generated from the following file:

- `artdaq_mfextensions/mfextensions/Destinations/SMTP_mfPlugin.cc`

## 4.13 mfplugins::ELTRACE Class Reference

Message Facility destination which logs messages to a TRACE buffer

Inheritance diagram for mfplugins::ELTRACE:



## Classes

- struct [Config](#)  
*Configuration Parameters for [ELTRACE](#).*

## Public Types

- using [Parameters](#) = fhicl::WrappedTable< [Config](#) >  
*Used for ParameterSet validation.*

## Public Member Functions

- [ELTRACE](#) ([Parameters](#) const &pset)  
*[ELTRACE](#) Constructor*

- void [fillPrefix](#) (std::ostream &o, const ErrorObj &msg) override  
*Fill the "Prefix" portion of the message.*
- void [fillUsrMsg](#) (std::ostream &o, const ErrorObj &msg) override  
*Fill the "User Message" portion of the message.*
- void [fillSuffix](#) (std::ostream &, const ErrorObj &) override  
*Fill the "Suffix" portion of the message (Unused)*
- void [routePayload](#) (const std::ostream &o, const ErrorObj &msg) override  
*Serialize a MessageFacility message to the output.*

### 4.13.1 Detailed Description

Message Facility destination which logs messages to a TRACE buffer

Definition at line 32 of file TRACE\_mfPlugin.cc.

### 4.13.2 Constructor & Destructor Documentation

#### 4.13.2.1 mfplugins::ELTRACE::ELTRACE ( Parameters const & pset )

[ELTRACE](#) Constructor

Parameters

<i>pset</i>	ParameterSet used to configure <a href="#">ELTRACE</a>
-------------	--

Definition at line 93 of file TRACE\_mfPlugin.cc.

### 4.13.3 Member Function Documentation

#### 4.13.3.1 void mfplugins::ELTRACE::fillPrefix ( std::ostream & o, const ErrorObj & msg ) [override]

Fill the "Prefix" portion of the message.

Parameters

<i>o</i>	Output stringstream
<i>msg</i>	MessageFacility object containing header information

Definition at line 115 of file TRACE\_mfPlugin.cc.

#### 4.13.3.2 void mfplugins::ELTRACE::fillUsrMsg ( std::ostream & o, const ErrorObj & msg ) [override]

Fill the "User Message" portion of the message.

Parameters

<i>o</i>	Output stringstream
<i>msg</i>	MessageFacility object containing header information

Definition at line 128 of file TRACE\_mfPlugin.cc.

4.13.3.3 `void mfplugins::ELTRACE::routePayload ( const std::ostream & o, const ErrorObj & msg )` [override]

Serialize a MessageFacility message to the output.



## Parameters

<i>o</i>	Stringstream object containing message data
<i>msg</i>	MessageFacility object containing header information

Definition at line 142 of file TRACE\_mfPlugin.cc.

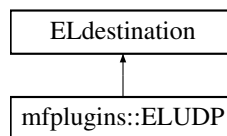
The documentation for this class was generated from the following file:

- artdaq\_mfextensions/mfextensions/Destinations/TRACE\_mfPlugin.cc

## 4.14 mfplugins::ELUDP Class Reference

Message Facility UDP Streamer Destination Formats messages into a delimited string and sends via UDP

Inheritance diagram for mfplugins::ELUDP:



### Classes

- struct [Config](#)

*Configuration Parameters for [ELUDP](#).*

### Public Types

- using [Parameters](#) = fhicl::WrappedTable< [Config](#) >

*Used for ParameterSet validation.*

### Public Member Functions

- [ELUDP](#) ([Parameters](#) const &pset)  
*[ELUDP](#) Constructor*
- void [fillPrefix](#) (std::ostream &o, const ErrorObj &msg) override  
*Fill the "Prefix" portion of the message.*
- void [fillUsrMsg](#) (std::ostream &o, const ErrorObj &msg) override  
*Fill the "User Message" portion of the message.*
- void [fillSuffix](#) (std::ostream &, const ErrorObj &) override  
*Fill the "Suffix" portion of the message (Unused)*
- void [routePayload](#) (const std::ostream &o, const ErrorObj &e) override  
*Serialize a MessageFacility message to the output.*

#### 4.14.1 Detailed Description

Message Facility UDP Streamer Destination Formats messages into a delimited string and sends via UDP

Definition at line 37 of file UDP\_mfPlugin.cc.

#### 4.14.2 Constructor & Destructor Documentation

##### 4.14.2.1 mfplugins::ELUDP::ELUDP ( Parameters const & *pset* )

[ELUDP](#) Constructor

Parameters

<i>pset</i>	ParameterSet used to configure <a href="#">ELUDP</a>
-------------	--

Definition at line 144 of file UDP\_mfPlugin.cc.

#### 4.14.3 Member Function Documentation

##### 4.14.3.1 void mfplugins::ELUDP::fillPrefix ( std::ostream & *o*, const ErrorObj & *msg* ) [override]

Fill the "Prefix" portion of the message.

Parameters

<i>o</i>	Output stringstream
<i>msg</i>	MessageFacility object containing header information

Definition at line 311 of file UDP\_mfPlugin.cc.

##### 4.14.3.2 void mfplugins::ELUDP::fillUsrMsg ( std::ostream & *o*, const ErrorObj & *msg* ) [override]

Fill the "User Message" portion of the message.

Parameters

<i>o</i>	Output stringstream
<i>msg</i>	MessageFacility object containing header information

Definition at line 363 of file UDP\_mfPlugin.cc.

##### 4.14.3.3 void mfplugins::ELUDP::routePayload ( const std::ostream & *o*, const ErrorObj & *e* ) [override]

Serialize a MessageFacility message to the output.

Parameters

<i>o</i>	Stringstream object containing message data
<i>e</i>	MessageFacility object containing header information

Definition at line 381 of file UDP\_mfPlugin.cc.

The documentation for this class was generated from the following file:

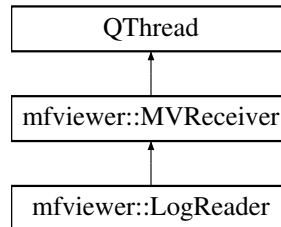
- artdaq\_mfextensions/mfextensions/Destinations/UDP\_mfPlugin.cc

## 4.15 mfviewer::LogReader Class Reference

MessageFacility Log Reader Read messagefacility log archive and reemit as messagefacility messages

```
#include <mfextensions/Receivers/LogReader_receiver.hh>
```

Inheritance diagram for mfviewer::LogReader:



### Public Member Functions

- [LogReader](#) (const fhicl::ParameterSet &pset)  
*LogReader Constructor*
- virtual [~LogReader](#) ()  
*LogReader Destructor*
- void [run](#) ()  
*Receiver loop method. Reads messages from file and emits newMessage signal*
- msg\_ptr\_t [read\\_next](#) ()  
*Read the next message from the input stream*
- bool [iseof](#) ()  
*Determine if the LogReader has reached the end of file.*

### Additional Inherited Members

#### 4.15.1 Detailed Description

MessageFacility Log Reader Read messagefacility log archive and reemit as messagefacility messages

Definition at line 20 of file LogReader\_receiver.hh.

#### 4.15.2 Constructor & Destructor Documentation

4.15.2.1 mfviewer::LogReader::LogReader ( const fhicl::ParameterSet &pset ) [explicit]

[LogReader](#) Constructor

Parameters

<i>pset</i>	ParameterSet used to configure the <a href="#">LogReader</a>
-------------	--

Definition at line 5 of file LogReader\_receiver.cc.

#### 4.15.2.2 `mfviewer::LogReader::~~LogReader ( )` [virtual]

[LogReader](#) Destructor

Definition at line 14 of file `LogReader_receiver.cc`.

### 4.15.3 Member Function Documentation

#### 4.15.3.1 `bool mfviewer::LogReader::iseof ( )` [inline]

Determine if the [LogReader](#) has reached the end of file.

##### Returns

Whether the reader is at the end of the file

Definition at line 50 of file `LogReader_receiver.hh`.

#### 4.15.3.2 `msg_ptr_t mfviewer::LogReader::read_next ( )`

Read the next message from the input stream

##### Returns

[qt\\_mf\\_msg](#) from log file

Definition at line 75 of file `LogReader_receiver.cc`.

#### 4.15.3.3 `void mfviewer::LogReader::run ( )`

Receiver loop method. Reads messages from file and emits `newMessage` signal

Definition at line 26 of file `LogReader_receiver.cc`.

The documentation for this class was generated from the following files:

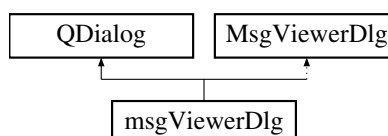
- `artdaq_mfextensions/mfextensions/Receivers/LogReader_receiver.hh`
- `artdaq_mfextensions/mfextensions/Receivers/LogReader_receiver.cc`

## 4.16 `msgViewerDlg` Class Reference

Message Viewer Dialog Window

```
#include <mfextensions/Binaries/mvdlg.hh>
```

Inheritance diagram for `msgViewerDlg`:



## Public Member Functions

- `template<typename T >`  
`void qt_check_for_QOBJECT_macro (const T &_q_argument) const`
- `virtual const QMetaObject * metaObject () const`
- `virtual void * qt_metacast (const char *)`
- `virtual int qt_metacall (QMetaObject::Call, int, void **)`
- `msgViewerDlg (std::string const &conf, QDialog *parent=nullptr)`  
*Message Viewer Dialog Constructor.*
- `void pause ()`  
*Pause message receiving.*
- `void exit ()`  
*Exit the program.*
- `void clear ()`  
*Clear the message buffer.*
- `void shortMode ()`  
*Switch to/from Short message mode.*
- `void changeSeverity (int sev)`  
*Change the severity threshold.*

## Static Public Member Functions

- `static QString tr (const char *s, const char *c=0)`
- `static QString trUtf8 (const char *s, const char *c=0)`
- `static QString tr (const char *s, const char *c, int n)`
- `static QString trUtf8 (const char *s, const char *c, int n)`

## Static Public Attributes

- `static const QMetaObject staticMetaObject`

## Protected Member Functions

- `void closeEvent (QCloseEvent *event)`  
*Perform actions on window close.*

### 4.16.1 Detailed Description

Message Viewer Dialog Window

Definition at line 27 of file mvdg.hh.

### 4.16.2 Constructor & Destructor Documentation

#### 4.16.2.1 msgViewerDlg::msgViewerDlg ( std::string const & conf, QDialog \* parent = nullptr )

Message Viewer Dialog Constructor.

## Parameters

<i>conf</i>	Configuration filename (fhicl document)
<i>parent</i>	Parent Qt window

Definition at line 83 of file mvdlg.cc.

### 4.16.3 Member Function Documentation

#### 4.16.3.1 void msgViewerDlg::closeEvent ( QCloseEvent \* *event* ) [protected]

Perform actions on window close.

## Parameters

<i>event</i>	QCloseEvent data
--------------	------------------

Definition at line 1102 of file mvdlg.cc.

The documentation for this class was generated from the following files:

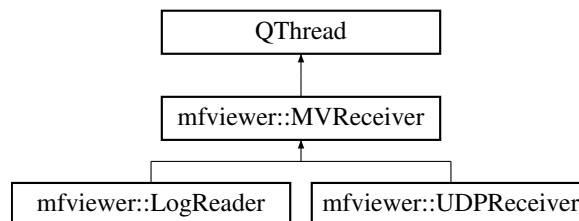
- artdaq\_mfextensions/mfextensions/Binaries/mvdlg.hh
- artdaq\_mfextensions/mfextensions/Binaries/mvdlg.cc

## 4.17 mfviewer::MVReceiver Class Reference

A [MVReceiver](#) class listens for messages and raises a signal when one arrives

```
#include <mfextensions/Receivers/MVReceiver.hh>
```

Inheritance diagram for mfviewer::MVReceiver:



### Public Member Functions

- template<typename T >  
void **qt\_check\_for\_QOBJECT\_macro** (const T &\_q\_argument) const
- virtual const QMetaObject \* **metaObject** () const
- virtual void \* **qt\_metacast** (const char \*)
- virtual int **qt\_metacall** (QMetaObject::Call, int, void \*\*)
- [MVReceiver](#) (fhicl::ParameterSet const &pset)  
*Construct a [MVReceiver](#) using the given ParameterSet*
- virtual [~MVReceiver](#) ()  
*[MVReceiver](#) destructor*
- void **stop** ()  
*Stop the [MVReceiver](#) thread*

## Static Public Member Functions

- static QString **tr** (const char \*s, const char \*c=0)
- static QString **trUtf8** (const char \*s, const char \*c=0)
- static QString **tr** (const char \*s, const char \*c, int n)
- static QString **trUtf8** (const char \*s, const char \*c, int n)

## Static Public Attributes

- static const QMetaObject **staticMetaObject**

## Protected Member Functions

- void [NewMessage](#) (msg\_ptr\_t const &msg)

*When a message is received by the [MVReceiver](#), this signal should be raised so that the connected listener can process it.*

## Protected Attributes

- std::atomic< bool > [stopRequested\\_](#)

*Whether the MVRecevier should stop*

### 4.17.1 Detailed Description

A [MVReceiver](#) class listens for messages and raises a signal when one arrives

Definition at line 17 of file MVReceiver.hh.

### 4.17.2 Constructor & Destructor Documentation

#### 4.17.2.1 mfviewer::MVReceiver::MVReceiver ( fhicl::ParameterSet const & pset ) [explicit]

Construct a [MVReceiver](#) using the given ParameterSet

Parameters

<i>pset</i>	ParameterSet used to construct <a href="#">MVReceiver</a>
-------------	---

Definition at line 3 of file MVReceiver.cc.

#### 4.17.2.2 virtual mfviewer::MVReceiver::~~MVReceiver ( ) [inline],[virtual]

[MVReceiver](#) destructor

Definition at line 31 of file MVReceiver.hh.

### 4.17.3 Member Function Documentation

4.17.3.1 `void mfviewer::MVReceiver::NewMessage ( msg_ptr_t const & msg )` `[protected]`

When a message is received by the [MVReceiver](#), this signal should be raised so that the connected listener can process it.



## Parameters

<i>msg</i>	Received message
------------	------------------

## 4.17.3.2 void mfviewer::MVReceiver::stop ( ) [inline]

Stop the [MVReceiver](#) thread

Definition at line 36 of file MVReceiver.hh.

## 4.17.4 Member Data Documentation

## 4.17.4.1 std::atomic&lt;bool&gt; mfviewer::MVReceiver::stopRequested\_ [protected]

Whether the MVReceiver should stop

Definition at line 42 of file MVReceiver.hh.

The documentation for this class was generated from the following files:

- artdaq\_mfextensions/mfextensions/Receivers/MVReceiver.hh
- artdaq\_mfextensions/mfextensions/Receivers/MVReceiver.cc

## 4.18 qt\_mf\_msg Class Reference

Qt wrapper around MessageFacility message

```
#include <mfextensions/Receivers/qt_mf_msg.hh>
```

## Public Member Functions

- [qt\\_mf\\_msg](#) (const std::string &hostname, const std::string &category, const std::string &application, pid\_t pid, timeval time)  
Construct a [qt\\_mf\\_msg](#)
- [qt\\_mf\\_msg](#) ()  
Default message constructor.
- [qt\\_mf\\_msg](#) (const [qt\\_mf\\_msg](#) &)=default  
Default copy constructor.
- [qt\\_mf\\_msg](#) ([qt\\_mf\\_msg](#) &&)=default  
Default Move Constructor.
- [qt\\_mf\\_msg](#) & operator= ([qt\\_mf\\_msg](#) const &)=default  
Default Copy Assignment Operator.
- [qt\\_mf\\_msg](#) & operator= ([qt\\_mf\\_msg](#) &&)=default  
Default Move Assignment Operator.
- virtual ~[qt\\_mf\\_msg](#) ()=default  
Default Destructor.
- QString const & [text](#) (bool mode) const  
Get the text of the message
- QColor const & [color](#) () const

- Get the severity-based color of the message*
- `sev_code_t sev () const`
- Get the severity of the message*
- `QString const & host () const`
- Get the host from which the message came*
- `QString const & cat () const`
- Get the category of the message*
- `QString const & app () const`
- Get the application of the message*
- `timeval time () const`
- Get the message timestamp*
- `size_t seq () const`
- Get the sequence number of the message*
- `void setSeverity (mf::ELseverityLevel sev)`
- Set the Severity of the message (MF levels)*
- `void setSeverityLevel (sev_code_t sev)`
- Set the severity code of the message (Viewer levels)*
- `void setMessage (const std::string &prefix, int iteration, const std::string &msg)`
- Set the message*
- `void setHostAddr (std::string const &hostaddr)`
- Set the hostaddr field*
- `void setFileName (std::string const &file)`
- Set the file name field*
- `void setLineNumber (std::string const &line)`
- Set the line number field*
- `void setModule (std::string const &module)`
- Set the module name*
- `void setEventID (std::string const &eventID)`
- Set the Event ID of the message*
- `void updateText ()`
- Parse fields and create HTML string representing message*

#### 4.18.1 Detailed Description

Qt wrapper around MessageFacility message

Definition at line 37 of file `qt_mf_msg.hh`.

#### 4.18.2 Constructor & Destructor Documentation

4.18.2.1 `qt_mf_msg::qt_mf_msg ( const std::string & hostname, const std::string & category, const std::string & application, pid_t pid, timeval time )`

Construct a `qt_mf_msg`

## Parameters

<i>hostname</i>	Hostname of the message source
<i>category</i>	Category of the message
<i>application</i>	Application of the message
<i>pid</i>	PID of the message
<i>time</i>	Timestamp of the message

Definition at line 10 of file qt\_mf\_msg.cc.

### 4.18.3 Member Function Documentation

#### 4.18.3.1 QString const& qt\_mf\_msg::app ( ) const [inline]

Get the application of the message

## Returns

Message application

Definition at line 91 of file qt\_mf\_msg.hh.

#### 4.18.3.2 QString const& qt\_mf\_msg::cat ( ) const [inline]

Get the category of the message

## Returns

Message category

Definition at line 86 of file qt\_mf\_msg.hh.

#### 4.18.3.3 QColor const& qt\_mf\_msg::color ( ) const [inline]

Get the severity-based color of the message

## Returns

Color of the message

Definition at line 71 of file qt\_mf\_msg.hh.

#### 4.18.3.4 QString const& qt\_mf\_msg::host ( ) const [inline]

Get the host from which the message came

## Returns

Hostname of message

Definition at line 81 of file qt\_mf\_msg.hh.

4.18.3.5 `size_t qt_mf_msg::seq ( ) const [inline]`

Get the sequence number of the message

Returns

Message sequence number

Definition at line 101 of file qt\_mf\_msg.hh.

4.18.3.6 `void qt_mf_msg::setEventID ( std::string const & eventID ) [inline]`

Set the Event ID of the message

Parameters

<i>eventID</i>	Event ID to set
----------------	-----------------

Definition at line 144 of file qt\_mf\_msg.hh.

4.18.3.7 `void qt_mf_msg::setFileName ( std::string const & file ) [inline]`

Set the file name field

Parameters

<i>file</i>	File generating message
-------------	-------------------------

Definition at line 129 of file qt\_mf\_msg.hh.

4.18.3.8 `void qt_mf_msg::setHostAddr ( std::string const & hostaddr ) [inline]`

Set the hostaddr field

Parameters

<i>hostaddr</i>	Host address of message source
-----------------	--------------------------------

Definition at line 124 of file qt\_mf\_msg.hh.

4.18.3.9 `void qt_mf_msg::setLineNumber ( std::string const & line ) [inline]`

Set the line number field

Parameters

<i>line</i>	Line number in file
-------------	---------------------

Definition at line 134 of file qt\_mf\_msg.hh.

4.18.3.10 `void qt_mf_msg::setMessage ( const std::string & prefix, int iteration, const std::string & msg )`

Set the message

## Parameters

<i>prefix</i>	Message prefix
<i>iteration</i>	Message iteration (run/event no)
<i>msg</i>	Message text

Definition at line 55 of file qt\_mf\_msg.cc.

4.18.3.11 `void qt_mf_msg::setModule ( std::string const & module ) [inline]`

Set the module name

## Parameters

<i>module</i>	Module generating message
---------------	---------------------------

Definition at line 139 of file qt\_mf\_msg.hh.

4.18.3.12 `void qt_mf_msg::setSeverity ( mf::ELseverityLevel sev )`

Set the Severity of the message (MF levels)

## Parameters

<i>sev</i>	Severity level of the message
------------	-------------------------------

Definition at line 24 of file qt\_mf\_msg.cc.

4.18.3.13 `void qt_mf_msg::setSeverityLevel ( sev_code_t sev ) [inline]`

Set the severity code of the message (Viewer levels)

## Parameters

<i>sev</i>	Severity code of the message
------------	------------------------------

Definition at line 112 of file qt\_mf\_msg.hh.

4.18.3.14 `sev_code_t qt_mf_msg::sev ( ) const [inline]`

Get the severity of the message

## Returns

Message severity

Definition at line 76 of file qt\_mf\_msg.hh.

4.18.3.15 `QString const& qt_mf_msg::text ( bool mode ) const [inline]`

Get the text of the message

## Parameters

<i>mode</i>	Whether to return the short-form text
-------------	---------------------------------------

## Returns

Text of the message

Definition at line 66 of file qt\_mf\_msg.hh.

**4.18.3.16** `timeval qt_mf_msg::time ( ) const` `[inline]`

Get the message timestamp

## Returns

Timestamp of the message

Definition at line 96 of file qt\_mf\_msg.hh.

**4.18.3.17** `void qt_mf_msg::updateText ( )`

Parse fields and create HTML string representing message

Definition at line 62 of file qt\_mf\_msg.cc.

The documentation for this class was generated from the following files:

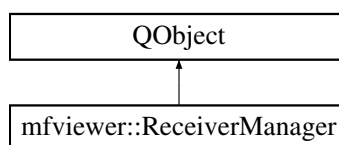
- artdaq\_mfextensions/mfextensions/Receivers/qt\_mf\_msg.hh
- artdaq\_mfextensions/mfextensions/Receivers/qt\_mf\_msg.cc

## 4.19 mfviewer::ReceiverManager Class Reference

The [ReceiverManager](#) loads one or more receiver plugins and displays messages received by those plugins on the Message Viewer dialog

```
#include <mfextensions/Receivers/ReceiverManager.hh>
```

Inheritance diagram for mfviewer::ReceiverManager:



## Signals

- void [newMessage](#) (msg\_ptr\_t const &msg)

*Signal raised on new message*

## Public Member Functions

- [ReceiverManager](#) (fhicl::ParameterSet const &pset)  
*ReceiverManager Constructor*
- virtual [~ReceiverManager](#) ()  
*ReceiverManager Destructor*
- void [start](#) ()  
*Start all receivers*
- void [stop](#) ()  
*Stop all receivers*

### 4.19.1 Detailed Description

The [ReceiverManager](#) loads one or more receiver plugins and displays messages received by those plugins on the Message Viewer dialog

Definition at line 13 of file ReceiverManager.hh.

### 4.19.2 Constructor & Destructor Documentation

4.19.2.1 mfviewer::ReceiverManager::ReceiverManager ( fhicl::ParameterSet const & pset ) [explicit]

[ReceiverManager](#) Constructor

Parameters

<i>pset</i>	ParameterSet used to configure <a href="#">ReceiverManager</a>
-------------	--

Definition at line 7 of file ReceiverManager.cc.

4.19.2.2 mfviewer::ReceiverManager::~~ReceiverManager ( ) [virtual]

[ReceiverManager](#) Destructor

Definition at line 31 of file ReceiverManager.cc.

### 4.19.3 Member Function Documentation

4.19.3.1 void mfviewer::ReceiverManager::newMessage ( msg\_ptr\_t const & msg ) [signal]

Signal raised on new message

Parameters

<i>msg</i>	Message just received
------------	-----------------------

4.19.3.2 void mfviewer::ReceiverManager::start ( )

Start all receivers

Definition at line 52 of file ReceiverManager.cc.

#### 4.19.3.3 void mfviewer::ReceiverManager::stop ( )

Stop all receivers

Definition at line 40 of file ReceiverManager.cc.

The documentation for this class was generated from the following files:

- artdaq\_mfextensions/mfextensions/Receivers/ReceiverManager.hh
- artdaq\_mfextensions/mfextensions/Receivers/ReceiverManager.cc

## 4.20 suppress Class Reference

Suppress messages based on a regular expression

```
#include <mfextensions/Extensions/suppress.hh>
```

### Public Member Functions

- [suppress](#) (std::string const &name)  
*Construct a suppression using the given name for regex matching*
- bool [match](#) (std::string const &name)  
*Check if the name matches this suppression*
- void [use](#) (bool flag)  
*Set whether the suppression is active*

#### 4.20.1 Detailed Description

Suppress messages based on a regular expression

Definition at line 13 of file suppress.hh.

#### 4.20.2 Constructor & Destructor Documentation

##### 4.20.2.1 suppress::suppress ( std::string const & name ) [explicit]

Construct a suppression using the given name for regex matching

Parameters

<i>name</i>	Name to suppress
-------------	------------------

Definition at line 3 of file suppress.cc.

#### 4.20.3 Member Function Documentation

##### 4.20.3.1 bool suppress::match ( std::string const & name )

Check if the name matches this suppression



## Parameters

<i>name</i>	Name to check
-------------	---------------

## Returns

True if name should be suppressed

Definition at line 6 of file suppress.cc.

4.20.3.2 void suppress::use ( bool *flag* ) [inline]

Set whether the suppression is active

## Parameters

<i>flag</i>	Whether the suppression should be active
-------------	--

Definition at line 33 of file suppress.hh.

The documentation for this class was generated from the following files:

- artdaq\_mfextensions/mfextensions/Extensions/suppress.hh
- artdaq\_mfextensions/mfextensions/Extensions/suppress.cc

## 4.21 throttle Class Reference

Throttle messages based on name and time limits. Separate from MessageFacility limiting.

```
#include <mfextensions/Extensions/throttle.hh>
```

### Public Member Functions

- [throttle](#) (std::string const &name, int limit, int64\_t timespan)  
*Throttle messages using a regular expression if they occur above a certain frequency*
- bool [reach\\_limit](#) (std::string const &name, timeval tm)  
*Determine whether the name has reached the throttling limit*
- void [use](#) (bool flag)  
*Enable or disable this throttle*

#### 4.21.1 Detailed Description

Throttle messages based on name and time limits. Separate from MessageFacility limiting.

Definition at line 17 of file throttle.hh.

#### 4.21.2 Constructor & Destructor Documentation

4.21.2.1 throttle::throttle ( std::string const & *name*, int *limit*, int64\_t *timespan* )

Throttle messages using a regular expression if they occur above a certain frequency

## Parameters

<i>name</i>	Regular expression to match messages
<i>limit</i>	Number of messages before throttling is enabled
<i>timespan</i>	Time limit for throttling

Definition at line 3 of file throttle.cc.

### 4.21.3 Member Function Documentation

#### 4.21.3.1 `bool throttle::reach_limit ( std::string const & name, timeval tm )`

Determine whether the name has reached the throttling limit

## Parameters

<i>name</i>	Name to check against regular expression
<i>tm</i>	Time of message

## Returns

Whether the message should be throttled

Definition at line 6 of file throttle.cc.

#### 4.21.3.2 `void throttle::use ( bool flag ) [inline]`

Enable or disable this throttle

## Parameters

<i>flag</i>	Whether the throttle should be enabled
-------------	--

Definition at line 40 of file throttle.hh.

The documentation for this class was generated from the following files:

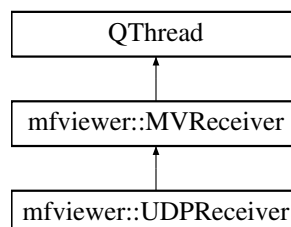
- `artdaq_mfextensions/mfextensions/Extensions/throttle.hh`
- `artdaq_mfextensions/mfextensions/Extensions/throttle.cc`

## 4.22 mfviewer::UDPReceiver Class Reference

Receive messages through a UDP socket. Expects the syslog format provided by UDP\_mfPlugin (ELUDP)

```
#include <mfextensions/Receivers/UDP_receiver.hh>
```

Inheritance diagram for `mfviewer::UDPReceiver`:



## Public Member Functions

- [UDPReceiver](#) (fhicl::ParameterSet const &pset)  
*UDPReceiver Constructor*
- virtual [~UDPReceiver](#) ()  
*Destructor – Close socket*
- void [run](#) () override  
*Receiver method. Receive messages and emit NewMessage signal*
- msg\_ptr\_t [read\\_msg](#) (std::string const &input)  
*Parse incoming message*

## Static Public Member Functions

- static bool [validate\\_packet](#) (std::string const &input)  
*Run simple validation tests on message*

## Additional Inherited Members

### 4.22.1 Detailed Description

Receive messages through a UDP socket. Expects the syslog format provided by UDP\_mfPlugin (ELUDP)  
Definition at line 12 of file UDP\_receiver.hh.

### 4.22.2 Constructor & Destructor Documentation

4.22.2.1 mfviewer::UDPReceiver::UDPReceiver ( fhicl::ParameterSet const & pset ) [explicit]

[UDPReceiver](#) Constructor

Parameters

<i>pset</i>	ParameterSet to use to configure the receiver
-------------	---

Definition at line 10 of file UDP\_receiver.cc.

4.22.2.2 mfviewer::UDPReceiver::~~UDPReceiver ( ) [virtual]

Destructor – Close socket

Definition at line 74 of file UDP\_receiver.cc.

### 4.22.3 Member Function Documentation

4.22.3.1 msg\_ptr\_t mfviewer::UDPReceiver::read\_msg ( std::string const & input )

Parse incoming message

**Parameters**

<i>input</i>	String to parse
--------------	-----------------

**Returns**

[qt\\_mf\\_msg](#) object containing message data

Definition at line 152 of file UDP\_receiver.cc.

#### 4.22.3.2 void mfviewer::UDPReceiver::run ( ) [override]

Receiver method. Receive messages and emit NewMessage signal

Definition at line 81 of file UDP\_receiver.cc.

#### 4.22.3.3 bool mfviewer::UDPReceiver::validate\_packet ( std::string const & *input* ) [static]

Run simple validation tests on message

**Parameters**

<i>input</i>	String to validate
--------------	--------------------

**Returns**

True if message contains "MF" marker and at least one "|" delimiter

Definition at line 290 of file UDP\_receiver.cc.

The documentation for this class was generated from the following files:

- artdaq\_mfextensions/mfextensions/Receivers/UDP\_receiver.hh
- artdaq\_mfextensions/mfextensions/Receivers/UDP\_receiver.cc

## 4.23 upload\_status Struct Reference

Structure to track progress of upload in cURL send function.

```
#include <mfextensions/Destinations/detail/curl_send_message.h>
```

**Public Attributes**

- size\_t [pos](#)  
*Current position within payload.*
- size\_t [size](#)  
*Size of payload.*
- const char \* [payload](#)  
*payload string*

#### 4.23.1 Detailed Description

Structure to track progress of upload in cURL send function.

Definition at line 21 of file curl\_send\_message.h.

The documentation for this struct was generated from the following file:

- [artdaq\\_mfextensions/mfextensions/Destinations/detail/curl\\_send\\_message.h](#)



## Chapter 5

# File Documentation

### 5.1 artdaq\_mfextensions/mfextensions/Destinations/detail/curl\_send\_message.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

#### Classes

- struct [upload\\_status](#)  
*Structure to track progress of upload in cURL send function.*

#### Functions

- void [send\\_message](#) (const char \*dest, const char \*to[], size\_t to\_size, const char \*from, const char \*payload, size\_t payload\_size)  
*Sends a message to the given SMTP server.*
- void [send\\_message\\_ssl](#) (const char \*dest, const char \*to[], size\_t to\_size, const char \*from, const char \*payload, size\_t payload\_size, const char \*username, const char \*pw, int disableVerify)  
*Sends a message to the given SMTP server, using SSL encryption.*

#### 5.1.1 Detailed Description

This file wraps the C-language cURL SMTP functions Code is from <https://curl.haxx.se/libcurl/c/example.-html>

Definition in file [curl\\_send\\_message.h](#).

#### 5.1.2 Function Documentation

5.1.2.1 `void send_message ( const char * dest, const char * to[], size_t to_size, const char * from, const char * payload, size_t payload_size )`

Sends a message to the given SMTP server.



## Parameters

<i>dest</i>	URL of SMTP server, in form smtp://[HOST]:[PORT]
<i>to</i>	Array of strings containing destination addresses
<i>to_size</i>	Size of the to array (must be >0!)
<i>from</i>	Address that the email is originating from
<i>payload</i>	Message payload, including RFC5322 headers
<i>payload_size</i>	Size of the message payload, in bytes

Definition at line 26 of file curl\_send\_message.c.

5.1.2.2 void send\_message\_ssl ( const char \* *dest*, const char \* *to*[], size\_t *to\_size*, const char \* *from*, const char \* *payload*, size\_t *payload\_size*, const char \* *username*, const char \* *pw*, int *disableVerify* )

Sends a message to the given SMTP server, using SSL encryption.

## Parameters

<i>dest</i>	URL of SMTP server, in form smtps://[HOST]:[PORT]
<i>to</i>	Array of strings containing destination addresses
<i>to_size</i>	Size of the to array (must be >0!)
<i>from</i>	Address that the email is originating from
<i>payload</i>	Message payload, including RFC5322 headers
<i>payload_size</i>	Size of the message payload, in bytes
<i>username</i>	Credentials for logging in to SMTPS server
<i>pw</i>	Credentials for logging in to SMTPS server (Recommend empty string)
<i>disableVerify</i>	Disable verification of host certificate (Recommend 0)

Definition at line 71 of file curl\_send\_message.c.

## 5.2 artdaq\_mfextensions/mfextensions/Receivers/detail/TCP\_listen\_fd.hh File Reference

```
#include <arpa/inet.h>
#include <errno.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <sys/socket.h>
#include "trace.h"
```

### Functions

- int [TCP\\_listen\\_fd](#) (int port, int rcvbuf)  
Create a TCP listening socket on the given port and INADDR\_ANY, with the given receive buffer.

### 5.2.1 Detailed Description

Defines a generator function for a TCP listen socket

Definition in file [TCP\\_listen\\_fd.hh](#).

## 5.2.2 Function Documentation

### 5.2.2.1 int TCP\_listen\_fd ( int *port*, int *rcvbuf* )

Create a TCP listening socket on the given port and INADDR\_ANY, with the given receive buffer.

#### Parameters

<i>port</i>	Port to listen on
<i>rcvbuf</i>	Receive buffer for socket. Set to 0 for TCP automatic buffer size

#### Returns

fd of new socket

Definition at line 32 of file [TCP\\_listen\\_fd.hh](#).

## 5.3 artdaq\_mfextensions/mfextensions/Receivers/detail/TCPConnect.hh File Reference

```
#include <arpa/inet.h>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <unistd.h>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <ifaddrs.h>
#include <linux/if_link.h>
#include <regex>
#include <string>
#include "TRACE/trace.h"
```

## Functions

- int [ResolveHost](#) (char const \*host\_in, in\_addr &addr)  
*Convert a string hostname to a in\_addr suitable for socket communication.*
- int [GetInterfaceForNetwork](#) (char const \*host\_in, in\_addr &addr)  
*Convert an IP address to the network address of the interface sharing the subnet mask.*
- int [ResolveHost](#) (char const \*host\_in, int dflt\_port, sockaddr\_in &sin)  
*Convert a string hostname and port to a sockaddr\_in suitable for socket communication.*
- int [TCPConnect](#) (char const \*host\_in, int dflt\_port, int64\_t flags=0, int sndbufsiz=0)  
*Connect to a host on a given port.*

### 5.3.1 Detailed Description

Provides utility functions for connecting TCP sockets

Definition in file [TCPConnect.hh](#).

### 5.3.2 Function Documentation

#### 5.3.2.1 `int GetInterfaceForNetwork ( char const * host_in, in_addr & addr ) [inline]`

Convert an IP address to the network address of the interface sharing the subnet mask.

Parameters

	<i>host_in</i>	IP to resolve
out	<i>addr</i>	in_addr object populated with resolved host

Returns

0 if success, -1 if gethostbyname fails, 2 if defaulted to 0.0.0.0 (No matching interfaces)

Definition at line 90 of file TCPConnect.hh.

#### 5.3.2.2 `int ResolveHost ( char const * host_in, in_addr & addr ) [inline]`

Convert a string hostname to a in\_addr suitable for socket communication.

Parameters

	<i>host_in</i>	Name or IP of host to resolve
out	<i>addr</i>	in_addr object populated with resolved host

Returns

0 if success, -1 if gethostbyname fails

Definition at line 42 of file TCPConnect.hh.

#### 5.3.2.3 `int ResolveHost ( char const * host_in, int dflt_port, sockaddr_in & sin ) [inline]`

Convert a string hostname and port to a sockaddr\_in suitable for socket communication.

Parameters

	<i>host_in</i>	Name or IP of host to resolve
	<i>dflt_port</i>	Port to populate in output
out	<i>sin</i>	sockaddr_in object populated with resolved host and port

Returns

0 if success, -1 if gethostbyname fails

Definition at line 184 of file TCPConnect.hh.

5.3.2.4 `int TCPConnect ( char const * host_in, int dflt_port, int64_t flags = 0, int sndbufsiz = 0 )` `[inline]`

Connect to a host on a given port.

#### Parameters

<i>host_in</i>	Name or IP of the host to connect to
<i>dflt_port</i>	Port to connect to
<i>flags</i>	TCP flags to use for the socket
<i>sndbufsiz</i>	Size of the send buffer. Set to 0 for automatic send buffer management

#### Returns

File descriptor of connected socket.

Definition at line 243 of file TCPConnect.hh.

# Index

- ~ELMultiFileOutput
  - mfplugins::ELMultiFileOutput, [23](#)
- ~LogReader
  - mfviewer::LogReader, [33](#)
- ~MVReceiver
  - mfviewer::MVReceiver, [37](#)
- ~ReceiverManager
  - mfviewer::ReceiverManager, [45](#)
- ~UDPReceiver
  - mfviewer::UDPReceiver, [49](#)
- app
  - qt\_mf\_msg, [41](#)
- append
  - mfplugins::ELMultiFileOutput::Config, [10](#)
- artdaq\_mfextensions/mfextensions/Destinations/detail/curl-  
\_send\_message.h, [53](#)
- artdaq\_mfextensions/mfextensions/Receivers/detail/TCP-  
\_listen\_fd.hh, [55](#)
- artdaq\_mfextensions/mfextensions/Receivers/detail/TCP-  
Connect.hh, [56](#)
- baseDir
  - mfplugins::ELMultiFileOutput::Config, [10](#)
- bellOnError
  - mfplugins::ELANSI::Config, [7](#)
- blinkOnError
  - mfplugins::ELANSI::Config, [8](#)
- cat
  - qt\_mf\_msg, [41](#)
- closeEvent
  - msgViewerDlg, [36](#)
- color
  - qt\_mf\_msg, [41](#)
- curl\_send\_message.h
  - send\_message, [53](#)
  - send\_message\_ssl, [55](#)
- debugColor
  - mfplugins::ELANSI::Config, [8](#)
- delimiter
  - mfplugins::ELFriendly::Config, [9](#)
- ELANSI
  - mfplugins::ELANSI, [19](#)
- ELFriendly
  - mfplugins::ELFriendly, [22](#)
- ELMultiFileOutput
  - mfplugins::ELMultiFileOutput, [23](#)
- ELOTS
  - mfplugins::ELOTS, [25](#)
- ELSMTP
  - mfplugins::ELSMTP, [27](#)
- ELTRACE
  - mfplugins::ELTRACE, [29](#)
- ELUDP
  - mfplugins::ELUDP, [32](#)
- error\_max
  - mfplugins::ELUDP::Config, [17](#)
- error\_report
  - mfplugins::ELUDP::Config, [17](#)
- errorColor
  - mfplugins::ELANSI::Config, [8](#)
- filename\_delimit
  - mfplugins::ELOTS::Config, [12](#)
  - mfplugins::ELUDP::Config, [17](#)
- fillPrefix
  - mfplugins::ELFriendly, [22](#)
  - mfplugins::ELOTS, [25](#)
  - mfplugins::ELTRACE, [29](#)
  - mfplugins::ELUDP, [32](#)
- fillSuffix
  - mfplugins::ELFriendly, [22](#)
- fillUsrMsg
  - mfplugins::ELFriendly, [22](#)
  - mfplugins::ELOTS, [25](#)
  - mfplugins::ELTRACE, [29](#)
  - mfplugins::ELUDP, [32](#)
- format\_string
  - mfplugins::ELOTS::Config, [12](#)
- from
  - mfplugins::ELSMTP::Config, [13](#)
- GetInterfaceForNetwork
  - TCPConnect.hh, [57](#)
- host
  - mfplugins::ELSMTP::Config, [13](#)
  - mfplugins::ELUDP::Config, [18](#)
  - qt\_mf\_msg, [41](#)

- infoColor
  - mfplugins::ELANSI::Config, 8
- iseof
  - mfviewer::LogReader, 34
- LogReader
  - mfviewer::LogReader, 33
- lvIm
  - mfplugins::ELTRACE::Config, 16
- lvls
  - mfplugins::ELTRACE::Config, 16
- MVReceiver
  - mfviewer::MVReceiver, 37
- match
  - suppress, 46
- messageHeader
  - mfplugins::ELSMTP::Config, 14
- mfplugins::ELANSI, 19
  - ELANSI, 19
  - routePayload, 20
- mfplugins::ELANSI::Config, 7
  - bellOnError, 7
  - blinkOnError, 8
  - debugColor, 8
  - errorColor, 8
  - infoColor, 8
  - warningColor, 8
- mfplugins::ELFriendly, 21
  - ELFriendly, 22
  - fillPrefix, 22
  - fillSuffix, 22
  - fillUsrMsg, 22
- mfplugins::ELFriendly::Config, 9
  - delimiter, 9
- mfplugins::ELMultiFileOutput, 22
  - ~ELMultiFileOutput, 23
  - ELMultiFileOutput, 23
  - routePayload, 24
- mfplugins::ELMultiFileOutput::Config, 10
  - append, 10
  - baseDir, 10
  - useApplication, 10
  - useCategory, 11
  - useHostname, 11
  - useModule, 11
- mfplugins::ELOTS, 24
  - ELOTS, 25
  - fillPrefix, 25
  - fillUsrMsg, 25
  - routePayload, 25
- mfplugins::ELOTS::Config, 11
  - filename\_delimit, 12
  - format\_string, 12
- mfplugins::ELSMTP, 26
  - ELSMTP, 27
  - routePayload, 28
- mfplugins::ELSMTP::Config, 12
  - from, 13
  - host, 13
  - messageHeader, 14
  - pw, 14
  - sendInterval, 14
  - subject, 14
  - to, 14
  - useSmtps, 15
  - user, 15
  - verifyCert, 15
- mfplugins::ELTRACE, 28
  - ELTRACE, 29
  - fillPrefix, 29
  - fillUsrMsg, 29
  - routePayload, 29
- mfplugins::ELTRACE::Config, 15
  - lvIm, 16
  - lvls, 16
- mfplugins::ELUDP, 31
  - ELUDP, 32
  - fillPrefix, 32
  - fillUsrMsg, 32
  - routePayload, 32
- mfplugins::ELUDP::Config, 16
  - error\_max, 17
  - error\_report, 17
  - filename\_delimit, 17
  - host, 18
  - multicast\_enabled, 18
  - output\_address, 18
- mfviewer::LogReader, 33
  - ~LogReader, 33
  - iseof, 34
  - LogReader, 33
  - read\_next, 34
  - run, 34
- mfviewer::MVReceiver, 36
  - ~MVReceiver, 37
  - MVReceiver, 37
  - NewMessage, 38
  - stop, 39
  - stopRequested\_, 39
- mfviewer::ReceiverManager, 44
  - ~ReceiverManager, 45
  - newMessage, 45
  - ReceiverManager, 45
  - start, 45
  - stop, 45
- mfviewer::UDPReceiver, 48
  - ~UDPReceiver, 49
  - read\_msg, 49

- run, 50
- UDPReceiver, 49
- validate\_packet, 50
- msgViewerDlg, 34
  - closeEvent, 36
  - msgViewerDlg, 35
  - msgViewerDlg, 35
- multicast\_enabled
  - mfplugins::ELUDP::Config, 18
- NewMessage
  - mfviewer::MVReceiver, 38
- newMessage
  - mfviewer::ReceiverManager, 45
- output\_address
  - mfplugins::ELUDP::Config, 18
- pw
  - mfplugins::ELSMTP::Config, 14
- qt\_mf\_msg, 39
  - app, 41
  - cat, 41
  - color, 41
  - host, 41
  - qt\_mf\_msg, 40
  - qt\_mf\_msg, 40
  - seq, 41
  - setEventID, 42
  - setFileName, 42
  - setHostAddr, 42
  - setLineNumber, 42
  - setMessage, 42
  - setModule, 43
  - setSeverity, 43
  - setSeverityLevel, 43
  - sev, 43
  - text, 43
  - time, 44
  - updateText, 44
- reach\_limit
  - throttle, 48
- read\_msg
  - mfviewer::UDPReceiver, 49
- read\_next
  - mfviewer::LogReader, 34
- ReceiverManager
  - mfviewer::ReceiverManager, 45
- ResolveHost
  - TCPConnect.hh, 57
- routePayload
  - mfplugins::ELANSI, 20
  - mfplugins::ELMultiFileOutput, 24
- mfplugins::ELOTS, 25
- mfplugins::ELSMTP, 28
- mfplugins::ELTRACE, 29
- mfplugins::ELUDP, 32
- run
  - mfviewer::LogReader, 34
  - mfviewer::UDPReceiver, 50
- send\_message
  - curl\_send\_message.h, 53
- send\_message\_ssl
  - curl\_send\_message.h, 55
- sendInterval
  - mfplugins::ELSMTP::Config, 14
- seq
  - qt\_mf\_msg, 41
- setEventID
  - qt\_mf\_msg, 42
- setFileName
  - qt\_mf\_msg, 42
- setHostAddr
  - qt\_mf\_msg, 42
- setLineNumber
  - qt\_mf\_msg, 42
- setMessage
  - qt\_mf\_msg, 42
- setModule
  - qt\_mf\_msg, 43
- setSeverity
  - qt\_mf\_msg, 43
- setSeverityLevel
  - qt\_mf\_msg, 43
- sev
  - qt\_mf\_msg, 43
- start
  - mfviewer::ReceiverManager, 45
- stop
  - mfviewer::MVReceiver, 39
  - mfviewer::ReceiverManager, 45
- stopRequested\_
  - mfviewer::MVReceiver, 39
- subject
  - mfplugins::ELSMTP::Config, 14
- suppress, 46
  - match, 46
  - suppress, 46
  - use, 47
- TCP\_listen\_fd
  - TCP\_listen\_fd.hh, 56
- TCP\_listen\_fd.hh
  - TCP\_listen\_fd, 56
- TCPConnect
  - TCPConnect.hh, 57
- TCPConnect.hh



- GetInterfaceForNetwork, [57](#)
- ResolveHost, [57](#)
- TCPConnect, [57](#)
- text
  - qt\_mf\_msg, [43](#)
- throttle, [47](#)
  - reach\_limit, [48](#)
  - throttle, [47](#)
  - use, [48](#)
- time
  - qt\_mf\_msg, [44](#)
- to
  - mfplugins::ELSMTP::Config, [14](#)
- UDPReceiver
  - mfviewer::UDPReceiver, [49](#)
- updateText
  - qt\_mf\_msg, [44](#)
- upload\_status, [50](#)
- use
  - suppress, [47](#)
  - throttle, [48](#)
- useApplication
  - mfplugins::ELMultiFileOutput::Config, [10](#)
- useCategory
  - mfplugins::ELMultiFileOutput::Config, [11](#)
- useHostname
  - mfplugins::ELMultiFileOutput::Config, [11](#)
- useModule
  - mfplugins::ELMultiFileOutput::Config, [11](#)
- useSmtps
  - mfplugins::ELSMTP::Config, [15](#)
- user
  - mfplugins::ELSMTP::Config, [15](#)
- validate\_packet
  - mfviewer::UDPReceiver, [50](#)
- verifyCert
  - mfplugins::ELSMTP::Config, [15](#)
- warningColor
  - mfplugins::ELANSI::Config, [8](#)