



MÄK Data Logger 3.7 DIS Class Documentation

This file documents Logger API and Logger File API classes. It supports Logger 3.7. It is based on the header files and includes many of the comments from them. It provides an alternative to looking through the individual header files when you need information about constructors or other aspects of a class's functioning. We hope you find this format helpful.

Classes are listed alphabetically, in a class hierarchy, and by various components. The hypertext links will help you quickly jump to related information.

- [Class Hierarchy](#)
- [Alphabetical List](#)
- [Compound List](#)
- [File List](#)
- [Compound Members](#)
- [File Members](#)

Notes

- There are two separate versions of Logger API class documentation. One is for DIS and the other is for HLA.
- If you find a difference between the information in this file and the original header file shipped with your version of VR-Link, rely on the header file.

MÄK Technologies, Inc.
185 Alewife Brook Parkway
Cambridge, MA 02138
Phone: 617.876.8085
Fax: 617.876.9208
www.mak.com
support@mak.com

Logger 3.7 DIS Classes Class Hierarchy

[Go to the graphical class hierarchy](#)

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

- **DtAnnotationManager**
- **DtLgrInterface**
 - **DtLgrControlListener**
 - **DtMotifGui**
- **DtLgrScalingClock**
 - **DtLgrPbClock**
 - **DtLgrRecClock**
- **DtLgrState**
- **DtLogger**
- **DtLoggerStage**
 - **DtActionObj**
 - **DtFilterObj**
 - **DtSendObj**
 - **DtDisSimulate**
 - **DtSendToFile**
 - **DtSourceObj**
 - **DtDisSource**
 - **DtFileSource**
- **DtNote**
- **LtLoggerFile**
 - **LtLoggerPlaybackFile**
 - **LtLoggerRecordFile**
- **LtNetInteractionRecord**
- **LtNetPacketHeader**
- **LtNetPacketTail**
- **LtNetRemovalRecord**
- **LtNetUpdateRecord**
- **LtPacketInfo**

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

Logger 3.7 DIS Classes Compound Index

D	DtFilterObj	DtLgrState	DtSendToFile	LtNetInteractionRecord
DtActionObj	DtLgrControlListener	DtLogger	DtSourceObj	LtNetPacketHeader
DtAnnotationManager	DtLgrInterface	DtLoggerStage	L	LtNetPacketTail
DtDisSimulate	DtLgrPbClock	DtMotifGui	LtLoggerFile	LtNetRemovalRecord
DtDisSource	DtLgrRecClock	DtNote	LtLoggerPlaybackFile	LtNetUpdateRecord
DtFileSource	DtLgrScalingClock	DtSendObj	LtLoggerRecordFile	LtPacketInfo

Logger 3.7 DIS Classes Compound List

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

- **DtActionObj** (This is an abstract class that defines a action Object for the logger agent to perform specific actions. examples can be ramToools interface)
- **DtAnnotationManager**
- **DtDisSimulate**
- **DtDisSource**
- **DtFileSource**
- **DtFilterObj** (This is an abstract class that defines a filter for the logger agent to manage data. examples can be ramToools interface)
- **DtLgrControlListener**
- **DtLgrInterface**
- **DtLgrPbClock**
- **DtLgrRecClock**
- **DtLgrScalingClock**
- **DtLgrState**
- **DtLogger**
- **DtLoggerStage** (This is an abstract class that defines a logger pipeline Object for the logger agent to perform specific actions. examples can be ramToools interface)
- **DtMotifGui**
- **DtNote**
- **DtSendObj** (This is an abstract class that defines a target for the logger agent to send data. examples can be files, databases or a hla federation)
- **DtSendToFile** (For DT_HAVE_UNISTD_H this class will send a packet to a file)
- **DtSourceObj** (This is an abstract class that defines a data source for a logger agent. A source could be a HLA connection, TCP connection, etc. Each connection will override these virtual functions and produce conforming data to enter the data stream)
- **LtLoggerFile** (For DT_HAVE_UNISTD_H)
- **LtLoggerPlaybackFile**
- **LtLoggerRecordFile**
- **LtNetInteractionRecord**
- **LtNetPacketHeader**
- **LtNetPacketTail**

- **LtNetRemovalRecord**
- **LtNetUpdateRecord**
- **LtPacketInfo**

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

Logger 3.7 DIS Classes File List

Here is a list of all files with brief descriptions:

- [actionObj.h](#)
- [annMgr.h](#)
- [defines.h](#)
- [disSim.h](#)
- [disSource.h](#)
- [fileSource.h](#)
- [filterObj.h](#)
- [hActiveSource.h](#)
- [hlaSim.h](#)
- [hlaSource.h](#)
- [hPassSource.h](#)
- [initMethods.h](#)
- [lgrClock.h](#)
- [lgrCtrlListener.h](#)
- [lgrInterface.h](#)
- [lgrState.h](#)
- [lgrUtil.h](#)
- [logger.h](#)
- [loggerFile.h](#)
- [loggerPbFile.h](#)
- [loggerRecFile.h](#)
- [loggerStage.h](#)
- [motifGui.h](#)
- [netPacket.h](#)
- [note.h](#)
- [packetInfo.h](#)
- [sendObj.h](#)
- [sendToFile.h](#)
- [sourceObj.h](#)
- [stateInfo.h](#)

- **winGui.h**

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

Logger 3.7 DIS Classes Compound Members

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [w](#) | [~](#)

Here is a list of all class members with links to the class documentation for each member:

- a -

- `aboutCallback()` : [DtMotifGui](#)
- `aboutOkCallback()` : [DtMotifGui](#)
- `aboutWidget` : [DtMotifGui](#)
- `activate()` : [DtActionObj](#), [DtSourceObj](#), [DtSendToFile](#), [DtSendObj](#), [DtLoggerStage](#), [DtFilterObj](#), [DtFileSource](#), [DtDisSource](#), [DtDisSimulate](#)
- `actOnPacket()` : [DtActionObj](#)
- `add()` : [DtAnnotationManager](#)
- `addAnnotHelpCallback()` : [DtMotifGui](#)
- `addAnnotOkCallback()` : [DtMotifGui](#)
- `addAnnotTextWidget` : [DtMotifGui](#)
- `addAnnotTimeLabel` : [DtMotifGui](#)
- `addAnnotWidget` : [DtMotifGui](#)
- `addBrowseWidget` : [DtMotifGui](#)
- `addInputButton` : [DtMotifGui](#)
- `addInputCallback()` : [DtMotifGui](#)
- `addInputOkCallback()` : [DtMotifGui](#)
- `addInputSelectCallback()` : [DtMotifGui](#)
- `addInputWidget` : [DtMotifGui](#)
- `addObject()` : [DtLogger](#)
- `addOutputBrowseWidget` : [DtMotifGui](#)
- `addOutputButton` : [DtMotifGui](#)
- `addOutputCallback()` : [DtMotifGui](#)
- `addOutputOkCallback()` : [DtMotifGui](#)

- addOutputSelectCallback() : **DtMotifGui**
- addOutputTextWidget : **DtMotifGui**
- addOutputWidget : **DtMotifGui**
- addParmTextWidget : **DtMotifGui**
- allowingFileDeletion() : **DtLgrControlListener**
- annotAddCallback() : **DtMotifGui**
- annotations() : **DtLogger, DtLgrState**
- annotCallback() : **DtMotifGui**
- annotClearCallback() : **DtMotifGui**
- annotCloseCallback() : **DtMotifGui**
- annotCurFile : **DtMotifGui**
- annotGuiMapping : **DtMotifGui**
- annotHelpMenuCallback() : **DtMotifGui**
- annotImportCallback() : **DtMotifGui**
- annotList : **DtMotifGui**
- annotMenu : **DtMotifGui**
- annotModCallback() : **DtMotifGui**
- annotOpenCallback() : **DtMotifGui**
- annotRemoveCallback() : **DtMotifGui**
- annotSaveasCallback() : **DtMotifGui**
- annotSaveCallback() : **DtMotifGui**
- annotText : **DtMotifGui**
- annotWidget : **DtMotifGui**
- arrangeForMode() : **DtMotifGui**
- arrivalTime : **LtPacketInfo, LtNetPacketHeader**
- askYN() : **DtMotifGui**
- atEnd() : **LtLoggerPlaybackFile**
- atStart() : **LtLoggerPlaybackFile**

- b -

- bmArray : **DtMotifGui**
- browserOpen : **DtMotifGui**
- buttonActive : **DtMotifGui**
- buttonPressed : **DtMotifGui**

- C -

- cachedMode : **DtMotifGui**
- cachedScale : **DtMotifGui**
- cachedTime : **DtMotifGui**
- checkMagicNumber() : **LtLoggerPlaybackFile**
- classHandle : **LtNetInteractionRecord, LtNetRemovalRecord, LtNetUpdateRecord**
- clearFlags() : **DtLgrState**
- clock() : **DtLgrState**
- closeFile() : **LtLoggerFile**
- colorWidget() : **DtMotifGui**
- configCallback() : **DtMotifGui**
- configCancelCallback() : **DtMotifGui**
- configHelpCallback() : **DtMotifGui**
- configOkCallback() : **DtMotifGui**
- configRadioCallback() : **DtMotifGui**
- configWidget : **DtMotifGui**
- context : **DtMotifGui**
- count() : **DtAnnotationManager**
- create() : **DtSendToFile, DtFileSource, DtDisSource, DtDisSimulate**
- createInterface() : **DtMotifGui**
- currentFile() : **DtLogger**
- currentFileField : **DtMotifGui**
- currentStatusField : **DtMotifGui**
- currentTimeField : **DtMotifGui**
- currentTimeLabel : **DtMotifGui**

- d -

- dataScaling() : **DtLogger, DtLgrState**
- deactivate() : **DtActionObj, DtSourceObj, DtSendToFile, DtSendObj, DtLoggerStage, DtFilterObj, DtFileSource, DtDisSource, DtDisSimulate**
- deleteAllObjs() : **DtDisSimulate**
- displayButtons() : **DtMotifGui**
- DtActionObj() : **DtActionObj**
- DtAnnotationManager() : **DtAnnotationManager**
- DtDisSimulate() : **DtDisSimulate**
- DtDisSource() : **DtDisSource**
- DtFileSource() : **DtFileSource**
- DtFilterObj() : **DtFilterObj**

- DtLgrControlListener() : **DtLgrControlListener**
- DtLgrInterface() : **DtLgrInterface**
- DtLgrPbClock() : **DtLgrPbClock**
- DtLgrRecClock() : **DtLgrRecClock**
- DtLgrScalingClock() : **DtLgrScalingClock**
- DtLgrState() : **DtLgrState**
- DtLogger() : **DtLogger**
- DtLoggerStage() : **DtLoggerStage**
- DtMotifGui() : **DtMotifGui**
- DtNote() : **DtNote**
- DtSendObj() : **DtSendObj**
- DtSendToFile() : **DtSendToFile**
- DtSourceObj() : **DtSourceObj**

- e -

- editPix : **DtMotifGui**
- empty() : **DtAnnotationManager**
- eofAction() : **DtLogger**, **DtLgrState**
- eofActLoopRadio : **DtMotifGui**
- eofActPauseRadio : **DtMotifGui**
- eofActStopRadio : **DtMotifGui**
- EOFActToggleCallback() : **DtMotifGui**
- EOFButton : **DtMotifGui**
- EOFCallback() : **DtMotifGui**
- EOFLabel : **DtMotifGui**
- eofPix : **DtMotifGui**
- eofPixG : **DtMotifGui**
- exerciseConn() : **DtLogger**, **DtLgrState**

- f -

- fetchFromClipboard() : **DtLogger**, **DtLgrState**
- file() : **DtLogger**, **DtLgrState**
- fileBrowseCallback() : **DtMotifGui**
- fileBrowseOkCallback() : **DtMotifGui**
- fileBrowseWidget : **DtMotifGui**
- fileLength() : **DtLogger**

- fileMenu : **DtMotifGui**
- filePosition() : **DtLogger**, **DtLgrState**
- filePtr() : **LtLoggerFile**
- fileTimeBackwards() : **LtLoggerPlaybackFile**
- fileTimeForwards() : **LtLoggerPlaybackFile**
- filterPacket() : **DtFilterObj**
- first() : **DtAnnotationManager**
- firstPacketTime() : **LtLoggerPlaybackFile**
- forceUpdatesOnAll() : **DtDisSimulate**

- g -

- generatePacket() : **DtSourceObj**, **DtFileSource**, **DtDisSource**
- generatePackets() : **DtSourceObj**, **DtFileSource**, **DtDisSource**
- get() : **DtAnnotationManager**
- getConstraint() : **DtLgrPbClock**
- getConstScaledTime() : **DtLgrPbClock**
- getIdealScaledTime() : **DtLgrScalingClock**
- getName() : **DtActionObj**, **DtLoggerStage**, **DtFilterObj**
- getRealTime() : **DtLgrScalingClock**
- getTimeScale() : **DtLgrScalingClock**
- getTsIndex() : **DtMotifGui**
- gotoEnd() : **DtLogger**
- gotoStart() : **DtLogger**

- h -

- heartBeat() : **DtLogger**, **DtLgrState**
- heartbeatWidget : **DtMotifGui**
- helpCallback() : **DtMotifGui**
- helpMenu : **DtMotifGui**

- i -

- id() : **DtLgrControlListener**
- idealOffset : **DtLgrScalingClock**
- idealScaledToReal() : **DtLgrScalingClock**
- idealTime() : **DtLogger**

- importAnnotOkCallback() : **DtMotifGui**
- importAnnotWidget : **DtMotifGui**
- init() : **DtLogger**
- initSelf() : **DtActionObj, DtSourceObj, DtSendToFile, DtSendObj, DtMotifGui, DtLoggerStage, DtFilterObj, DtFileSource, DtDisSource, DtDisSimulate**
- inPlayback() : **DtLogger**
- inputList : **DtMotifGui**
- inputRemoveCallback() : **DtMotifGui**
- inputSelNum : **DtMotifGui**
- isOk() : **DtActionObj, DtSourceObj, DtSendObj, DtLoggerStage, LtLoggerFile, DtFilterObj**

- j -

- jumpCallback() : **DtMotifGui**
- jumpPix : **DtMotifGui**
- jumpToEnd() : **LtLoggerPlaybackFile**
- jumpToStart() : **LtLoggerPlaybackFile**

- l -

- lastPacketTime() : **LtLoggerPlaybackFile**
- lgrControlCb() : **DtLgrControlListener**
- listSelectionCallback() : **DtMotifGui**
- loadFileForPlayback() : **DtLogger**
- loadFileForRecording() : **DtLogger**
- loadFromFile() : **DtAnnotationManager**
- logoPix : **DtMotifGui**
- LtLoggerFile() : **LtLoggerFile**
- LtLoggerPlaybackFile() : **LtLoggerPlaybackFile**
- LtLoggerRecordFile() : **LtLoggerRecordFile**
- LtPacketInfo() : **LtPacketInfo**

- m -

- manInputCallback() : **DtMotifGui**
- manInputMenu : **DtMotifGui**
- manInputWidget : **DtMotifGui**
- manOutputCallback() : **DtMotifGui**

- manOutputMenu : **DtMotifGui**
- manOutputWidget : **DtMotifGui**
- menuBar : **DtMotifGui**
- modAnnotHelpCallback() : **DtMotifGui**
- modAnnotOkCallback() : **DtMotifGui**
- modAnnotTextWidget : **DtMotifGui**
- modAnnotTimeLabel : **DtMotifGui**
- modAnnotWidget : **DtMotifGui**
- mode() : **DtLogger**, **DtLgrState**
- msg() : **DtNote**
- myActive : **DtDisSource**
- myAdvanced : **DtMotifGui**
- myAllowFileDelete : **DtLgrControlListener**
- myAnnotAddTime : **DtMotifGui**
- myAnnotations : **DtLgrState**, **DtAnnotationManager**
- myAnnotCount : **DtMotifGui**
- myArrivalTime : **LtPacketInfo**
- myBrowser : **DtMotifGui**
- myClipboard : **DtLgrState**
- myClock : **DtLogger**, **DtLgrState**, **DtLgrScalingClock**
- myCreatedExConn : **DtLogger**
- myCurrent : **DtAnnotationManager**
- myDataScaling : **DtLgrState**
- myEndPos : **LtLoggerPlaybackFile**
- myEofAction : **DtLgrState**
- myExConn : **DtLogger**, **DtLgrState**
- myFile : **DtLgrState**
- myFileLength : **DtLgrState**
- myFilename : **DtLogger**
- myFileName : **DtSendToFile**, **LtLoggerFile**
- myFilePosition : **DtLgrState**
- myFirstTime : **LtLoggerPlaybackFile**
- myFp : **DtSendToFile**, **LtLoggerFile**
- myHeartbeat : **DtLogger**, **DtLgrState**
- myId : **DtLgrControlListener**
- myLast : **DtAnnotationManager**
- myLastTime : **LtLoggerPlaybackFile**
- myListenFlag : **DtDisSource**

- myLogger : **DtLgrInterface**
- myMax : **DtAnnotationManager**
- myMirror : **DtLogger**
- myMode : **DtLgrState**
- myName : **DtSourceObj, DtLoggerStage**
- myPacket : **LtPacketInfo**
- myPacketKind : **LtPacketInfo**
- myPacketLimit : **DtLgrState**
- myPacketList : **DtDisSource**
- myPb : **DtLogger**
- myPbSend : **DtLogger**
- myPbSource : **DtLogger**
- myPipeline : **DtLogger**
- myPlayAction : **DtLgrState**
- myPlaybackConstructMethod : **DtLgrState**
- myProcessAll : **DtLgrControlListener**
- myRecFile : **DtSendToFile**
- myRecLgrCtrlInfo : **DtLogger**
- myRecordAction : **DtLgrState**
- myRecordConstructMethod : **DtLgrState**
- myRecSend : **DtLogger**
- myRecSource : **DtLogger**
- myRecSrcParam : **DtLogger**
- mySelectedAnnotation : **DtMotifGui**
- mySize : **LtPacketInfo**
- mySkipToEnd : **DtLgrState**
- mySkipToStart : **DtLgrState**
- myStartPos : **LtLoggerPlaybackFile**
- myState : **DtSendToFile, DtLogger, DtDisSource**
- myStateRep : **DtLgrState**
- myStatus : **DtActionObj, DtSourceObj, DtSendObj, LtLoggerFile, DtFilterObj**
- myTimeJumped : **DtLgrState**
- myTimeout : **DtLogger, DtLgrState**
- myTimeScale : **DtLogger**

- n -

- name() : **LtLoggerFile**

- nameSize : [LtNetRemovalRecord](#), [LtNetUpdateRecord](#)
- newCallback() : [DtMotifGui](#)
- newFileCancelCallback() : [DtMotifGui](#)
- newFileOkCallback() : [DtMotifGui](#)
- newFileWidget : [DtMotifGui](#)
- newMenu : [DtMotifGui](#)
- next() : [DtAnnotationManager](#)
- numAttrs : [LtNetUpdateRecord](#)
- numBytes : [LtNetPacketTail](#), [LtNetPacketHeader](#)
- numParams : [LtNetInteractionRecord](#)
- numValueBytes : [LtNetInteractionRecord](#), [LtNetUpdateRecord](#)

- o -

- objectId : [LtNetRemovalRecord](#), [LtNetUpdateRecord](#)
- openActToggleBox : [DtMotifGui](#)
- openAnnotCancelCallback() : [DtMotifGui](#)
- openAnnotOkCallback() : [DtMotifGui](#)
- openAnnotWidget : [DtMotifGui](#)
- openCallback() : [DtMotifGui](#)
- openFile() : [LtLoggerFile](#)
- openFileCancelCallback() : [DtMotifGui](#)
- openFileOkCallback() : [DtMotifGui](#)
- openFileWidget : [DtMotifGui](#)
- openMenu : [DtMotifGui](#)
- operator=() : [LtPacketInfo](#)
- optionMenu : [DtMotifGui](#)
- outputList : [DtMotifGui](#)
- outputRemoveCallback() : [DtMotifGui](#)
- outputSelNum : [DtMotifGui](#)

- p -

- packet() : [LtPacketInfo](#)
- packetKind() : [LtPacketInfo](#)
- packetLimit() : [DtLgrState](#)
- pad1 : [LtNetRemovalRecord](#)
- pad2 : [LtNetRemovalRecord](#)

- padding : **LtNetUpdateRecord, LtNetPacketHeader**
- pause() : **DtLogger**
- pauseButton : **DtMotifGui**
- pauseCallback() : **DtMotifGui**
- pauseLabel : **DtMotifGui**
- pausePix : **DtMotifGui**
- pausePixG : **DtMotifGui**
- pausePixPG : **DtMotifGui**
- pb : **DtMotifGui**
- pendingEOFAction : **DtMotifGui**
- pendingPlayAction : **DtMotifGui**
- pendingRecordAction : **DtMotifGui**
- play() : **DtLogger**
- playAction() : **DtLogger, DtLgrState**
- playActPlayRadio : **DtMotifGui**
- playActStopRadio : **DtMotifGui**
- playActToggleCallback() : **DtMotifGui**
- playbackFileCreator() : **DtLgrState**
- playButton : **DtMotifGui**
- playCallback() : **DtMotifGui**
- playLabel : **DtMotifGui**
- playPix : **DtMotifGui**
- playPixG : **DtMotifGui**
- playPixP : **DtMotifGui**
- playPixPG : **DtMotifGui**
- playToggleBox : **DtMotifGui**
- postToClipboard() : **DtLogger, DtLgrState**
- printInfo() : **DtActionObj, DtSourceObj, DtSendToFile, DtSendObj, DtLoggerStage, DtFilterObj, DtFileSource, DtDisSource, DtDisSimulate**
- procAggPdu() : **DtDisSimulate**
- procEsPdu() : **DtDisSimulate**
- processAll() : **DtLgrControlListener**
- processAnnotationFile() : **DtMotifGui**
- processData() : **DtActionObj, DtSendToFile, DtSendObj, DtFilterObj, DtDisSimulate**
- processLgrControl() : **DtLgrControlListener**

- quitCallback() : **DtMotifGui**

- r -

- readPacket() : **LtPacketInfo, LtLoggerPlaybackFile**
- readPacketBackwards() : **LtLoggerPlaybackFile**
- readPacketData() : **LtLoggerPlaybackFile**
- readPacketHeader() : **LtLoggerPlaybackFile**
- readPacketTail() : **LtLoggerPlaybackFile**
- realToIdealScaled() : **DtLgrScalingClock**
- recClock : **DtDisSource**
- receivePdu() : **DtDisSource**
- receivePduCb() : **DtDisSource**
- record() : **DtLogger**
- recordAction() : **DtLogger, DtLgrState**
- recordActRecordRadio : **DtMotifGui**
- recordActStopRadio : **DtMotifGui**
- recordActToggleCallback() : **DtMotifGui**
- recordButton : **DtMotifGui**
- recordCallback() : **DtMotifGui**
- recordFileCreator() : **DtLgrState**
- recordingLoggerControlInfo() : **DtLogger**
- recordLabel : **DtMotifGui**
- recordPix : **DtMotifGui**
- recordPixG : **DtMotifGui**
- recordPixP : **DtMotifGui**
- recordPixPG : **DtMotifGui**
- registerObject() : **DtLogger**
- remove() : **DtAnnotationManager**
- removeFromClipboard() : **DtLogger, DtLgrState**
- removeInputButton : **DtMotifGui**
- removeObject() : **DtLogger**
- removeOutputButton : **DtMotifGui**
- reset() : **DtActionObj, DtSourceObj, DtSendObj, DtLoggerStage, DtLogger, DtFileSource, DtDisSource**
- resetAnnotList() : **DtMotifGui**

- S -

- saveasAnnotOkCallback() : **DtMotifGui**
- saveasAnnotWidget : **DtMotifGui**
- saveNoteList() : **DtMotifGui**
- saveToFile() : **DtAnnotationManager**
- seekBackward() : **LtLoggerPlaybackFile**
- seekForward() : **LtLoggerPlaybackFile**
- seekTime() : **LtLoggerPlaybackFile**
- send() : **DtDisSimulate**
- sendPacket() : **DtSendToFile, DtSendObj, DtDisSimulate**
- setAllowFileDeletion() : **DtLgrControlListener**
- setArrivalTime() : **LtPacketInfo**
- setConstraint() : **DtLgrPbClock**
- setConstScaledTime() : **DtLgrPbClock**
- setDataScaling() : **DtLogger, DtLgrState**
- setDefaultRecordSourceParameter() : **DtLogger**
- setEofAction() : **DtLogger, DtLgrState**
- setFile() : **DtLgrState**
- setFilePosition() : **DtLgrState**
- setHeartBeat() : **DtLogger, DtLgrState**
- setId() : **DtLgrControlListener**
- setIdealScaledTime() : **DtLgrScalingClock**
- setMirroring() : **DtLogger**
- setMode() : **DtLogger, DtLgrState**
- setMsg() : **DtNote**
- setPacket() : **LtPacketInfo**
- setPacketKind() : **LtPacketInfo**
- setPacketLimit() : **DtLogger, DtLgrState**
- setPlayAction() : **DtLogger, DtLgrState**
- setPlaybackFileCreator() : **DtLogger, DtLgrState**
- setProcessAll() : **DtLgrControlListener**
- setRecordAction() : **DtLogger, DtLgrState**
- setRecordFileCreator() : **DtLogger, DtLgrState**
- setRecordingLoggerControlInfo() : **DtLogger**
- setScaleCancelCallback() : **DtMotifGui**
- setScaleHelpCallback() : **DtMotifGui**
- setScaleOkCallback() : **DtMotifGui**
- setScaleWidget : **DtMotifGui**
- setSize() : **LtPacketInfo**

- setSkipToEnd() : **DtLgrState**
- setSkipToStart() : **DtLgrState**
- setTime() : **DtNote**, **DtLogger**
- setTimeCallback() : **DtMotifGui**
- setTimeCancelCallback() : **DtMotifGui**
- setTimeHelpCallback() : **DtMotifGui**
- setTimeJumped() : **DtLgrState**
- setTimeMenu : **DtMotifGui**
- setTimeOkCallback() : **DtMotifGui**
- setTimeout() : **DtLogger**, **DtLgrState**
- setTimeScale() : **DtLogger**, **DtLgrRecClock**, **DtLgrPbClock**, **DtLgrScalingClock**
- setTimeToQuit() : **DtLgrInterface**
- setTimeWidget : **DtMotifGui**
- setup() : **DtLogger**
- shadeAndSensitizeButton() : **DtMotifGui**
- shell : **DtMotifGui**
- showFilename() : **DtMotifGui**
- showMode() : **DtMotifGui**
- showPositional() : **DtMotifGui**
- showScale() : **DtMotifGui**
- showTime() : **DtMotifGui**
- size() : **LtPacketInfo**
- skipPacketBackwards() : **LtLoggerPlaybackFile**
- skipPacketForwards() : **LtLoggerPlaybackFile**
- skipTime() : **DtLogger**
- skipTimeCallback() : **DtMotifGui**
- skipTimeCancelCallback() : **DtMotifGui**
- skipTimeHelpCallback() : **DtMotifGui**
- skipTimeMenu : **DtMotifGui**
- skipTimeOkCallback() : **DtMotifGui**
- skipTimeToggleCallback() : **DtMotifGui**
- skipTimeWidget : **DtMotifGui**
- skipToEnd() : **DtLgrState**
- skipToStart() : **DtLgrState**
- SOFButton : **DtMotifGui**
- SOFCallback() : **DtMotifGui**
- SOFLabel : **DtMotifGui**
- sofPix : **DtMotifGui**

- sofPixG : **DtMotifGui**
- startListening() : **DtDisSource**
- stateReps() : **DtLgrState**
- stop() : **DtLogger**
- stopButton : **DtMotifGui**
- stopCallback() : **DtMotifGui**
- stopLabel : **DtMotifGui**
- stopPix : **DtMotifGui**
- stopPixG : **DtMotifGui**

- t -

- tick() : **DtActionObj, DtSourceObj, DtSendObj, DtMotifGui, DtLoggerStage, DtLogger, DtLgrInterface, DtFilterObj, DtFileSource, DtDisSimulate**
- time() : **DtNote, DtLogger**
- timeJumped() : **DtLgrState**
- timeout() : **DtLogger, DtLgrState**
- timeoutWidget : **DtMotifGui**
- timeScale : **DtLogger, DtLgrScalingClock**
- timeScaleCallback() : **DtMotifGui**
- timeScaleMenu : **DtMotifGui**
- timeScaleSlider : **DtMotifGui**
- timeScaleValues : **DtMotifGui**
- timeScaleValuesCount : **DtMotifGui**
- timeToQuit() : **DtLgrInterface**
- topForm : **DtMotifGui**
- topLevel : **DtMotifGui**
- topManager : **DtMotifGui**

- u -

- updateAnnotList() : **DtMotifGui**

- w -

- warn() : **DtMotifGui**
- writeMagicNumber() : **LtLoggerRecordFile**
- writePacket() : **LtPacketInfo, LtLoggerRecordFile**

- ~DtActionObj() : **DtActionObj**
- ~DtAnnotationManager() : **DtAnnotationManager**
- ~DtDisSimulate() : **DtDisSimulate**
- ~DtDisSource() : **DtDisSource**
- ~DtFileSource() : **DtFileSource**
- ~DtFilterObj() : **DtFilterObj**
- ~DtLgrControlListener() : **DtLgrControlListener**
- ~DtLgrInterface() : **DtLgrInterface**
- ~DtLgrPbClock() : **DtLgrPbClock**
- ~DtLgrRecClock() : **DtLgrRecClock**
- ~DtLgrScalingClock() : **DtLgrScalingClock**
- ~DtLgrState() : **DtLgrState**
- ~DtLogger() : **DtLogger**
- ~DtLoggerStage() : **DtLoggerStage**
- ~DtMotifGui() : **DtMotifGui**
- ~DtSendObj() : **DtSendObj**
- ~DtSendToFile() : **DtSendToFile**
- ~DtSourceObj() : **DtSourceObj**
- ~LtLoggerFile() : **LtLoggerFile**
- ~LtLoggerPlaybackFile() : **LtLoggerPlaybackFile**
- ~LtLoggerRecordFile() : **LtLoggerRecordFile**
- ~LtPacketInfo() : **LtPacketInfo**

Logger 3.7 DIS Classes File Members

[a](#) | [d](#) | [l](#) | [m](#)

Here is a list of all file members with links to the files they belong to:

- a -

- AGENT_LIB : [defines.h](#)

- d -

- DEFAULT_PACKET_LIMIT : [lgrState.h](#)
- DtCreateMethod : [initMethods.h](#)
- DtLoggerPlaybackConstructMethod : [initMethods.h](#)
- DtLoggerRecordConstructMethod : [initMethods.h](#)

- l -

- LGR_APP : [defines.h](#)
- LGR_CLOCK_FOREVER : [lgrClock.h](#)
- LGR_CLOCK_SMALL_SCALE : [lgrClock.h](#)
- LGR_EXPORT : [initMethods.h](#)
- LGR_LIB : [defines.h](#)
- LtAction : [defines.h](#)
- LtChopSlash() : [lgrUtil.h](#)
- LtFilter : [defines.h](#)
- LtMAX_PACKET : [packetInfo.h](#)
- LtObjectType : [defines.h](#)
- LtPacketAnnotation : [packetInfo.h](#)
- LtPacketAudio : [packetInfo.h](#)
- LtPacketFom : [packetInfo.h](#)
- LtPacketInteraction : [packetInfo.h](#)

- LtPacketKind : [packetInfo.h](#)
- LtPacketPdu : [packetInfo.h](#)
- LtPacketRemoval : [packetInfo.h](#)
- LtPacketUpdate : [packetInfo.h](#)
- LtSend : [defines.h](#)
- LtSource : [defines.h](#)
- LtStrEquiv() : [lgrUtil.h](#)
- LtStringToTime() : [lgrUtil.h](#)
- LtTimeToString() : [lgrUtil.h](#)
- LtToLower() : [lgrUtil.h](#)
- LtUndefined : [defines.h](#)

- m -

- MAX_ANNOTS : [motifGui.h](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

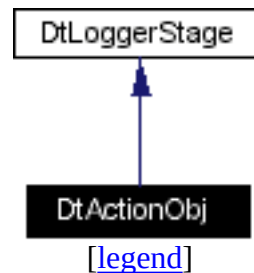
DtActionObj Class Reference

this is an abstract class that defines a action Object for the logger agent to perform specific actions.
examples can be ramToools interface. [More...](#)

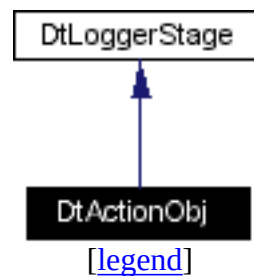
```
#include <actionObj.h>
```

Inherits **DtLoggerStage**.

Inheritance diagram for DtActionObj:



Collaboration diagram for DtActionObj:



[List of all members.](#)

Public Methods

- **DtActionObj** (const char *theName)
- virtual **~DtActionObj** ()
- virtual void **initSelf** (**DtLgrState** *theState)=0
this method is called once after construction to perform some initial initialization. [More...](#)
- virtual void **activate** ()=0

this method is called when the logger agent has selected to send data to this object. [More...](#)

- virtual void **deactivate** ()=0

this method is called when the logger agent is no longer going to send data to this object. [More...](#)

- virtual void **processData** (DtList *packetList)

this method will call the actOnPacket method for each packet in the packet list. [More...](#)

- virtual void **actOnPacket** (const **LtPacketInfo** *packet)=0

the actions are performed in this method. [More...](#)

- virtual void **tick** ()=0

this method gives the object execution time. [More...](#)

- virtual DtBoolean **isOk** () const

- virtual const char* **getName** () const

- virtual const char* **printInfo** () const=0

this method returns a string that supplies identifying info about the object. [More...](#)

- virtual void **reset** ()=0

Reset object. [More...](#)

Protected Attributes

- DtBoolean **myStatus**
-

Detailed Description

this is an abstract class that defines a action Object for the logger agent to perform specific actions.
examples can be ramToools interface.

Constructor & Destructor Documentation

```
DtActionObj::DtActionObj (const char * theName )
```

DtActionObj::~~DtActionObj () [virtual]

Member Function Documentation

void DtActionObj::actOnPacket (const [LtPacketInfo](#) * *packet*) [pure virtual]

the actions are performed in this method.

void DtActionObj::activate () [pure virtual]

this method is called when the logger agent has selected to send data to this object.

Reimplemented from [DtLoggerStage](#).

void DtActionObj::deactivate () [pure virtual]

this method is called when the logger agent is no longer going to send data to this object.

Reimplemented from [DtLoggerStage](#).

const char * DtActionObj::getName () const [inline, virtual]

Reimplemented from [DtLoggerStage](#).

void DtActionObj::initSelf ([DtLgrState](#) * *theState*) [pure virtual]

this method is called once after construction to perform some initial initialization.

Reimplemented from [DtLoggerStage](#).

DtBoolean DtActionObj::isOk () const [inline, virtual]

Reimplemented from [DtLoggerStage](#).

```
const char * DtActionObj::printInfo () const [pure virtual]
```

this method returns a string that supplies identifying info about the object.

Reimplemented from [DtLoggerStage](#).

```
void DtActionObj::processData (DtList * packetList ) [virtual]
```

this method will call the actOnPacket method for each packet in the packet list.

```
void DtActionObj::reset () [pure virtual]
```

Reset object.

Reimplemented from [DtLoggerStage](#).

```
void DtActionObj::tick () [pure virtual]
```

this method gives the object execution time.

Reimplemented from [DtLoggerStage](#).

Member Data Documentation

```
DtBoolean DtActionObj::myStatus [protected]
```

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

- [actionObj.h](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtAnnotationManager Class Reference

```
#include <annMgr.h>
```

Collaboration diagram for DtAnnotationManager:



[List of all members.](#)

Public Methods

- **DtAnnotationManager** (int maxAnnotations=100)
- virtual **~DtAnnotationManager** ()
- virtual void **add** (DtTime t, DtString name)
- virtual void **remove** (DtTime t)
- virtual int **count** () const
- virtual **DtNote*** **get** (DtTime t)
- virtual **DtNote*** **first** ()
- virtual **DtNote*** **next** ()
- virtual void **empty** ()
- virtual void **saveToFile** (const char *filename)
- virtual void **loadFromFile** (const char *filename)

Protected Attributes

- **DtNote**** **myAnnotations**
- int **myMax**

- int **myLast**
 - int **myCurrent**
-

Constructor & Destructor Documentation

```
DtAnnotationManager::DtAnnotationManager (int maxAnnotations = 100 )
```

```
DtAnnotationManager::~~DtAnnotationManager () [virtual]
```

Member Function Documentation

```
void DtAnnotationManager::add (DtTime t,  
                               DtString name ) [virtual]
```

```
int DtAnnotationManager::count () const [virtual]
```

```
void DtAnnotationManager::empty () [virtual]
```

```
DtNote * DtAnnotationManager::first () [virtual]
```

```
DtNote * DtAnnotationManager::get (DtTime t) [virtual]
```

```
void DtAnnotationManager::loadFromFile (const char * filename) [virtual]
```

```
DtNote * DtAnnotationManager::next () [virtual]
```

```
void DtAnnotationManager::remove (DtTime t) [virtual]
```

```
void DtAnnotationManager::saveToFile (const char * filename) [virtual]
```

Member Data Documentation

```
DtNote ** DtAnnotationManager::myAnnotations [protected]
```

```
int DtAnnotationManager::myCurrent [protected]
```

```
int DtAnnotationManager::myLast [protected]
```

```
int DtAnnotationManager::myMax [protected]
```

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

- [annMgr.h](#)

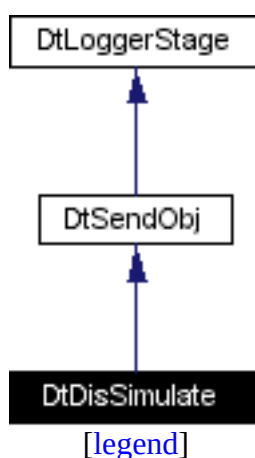
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtDisSimulate Class Reference

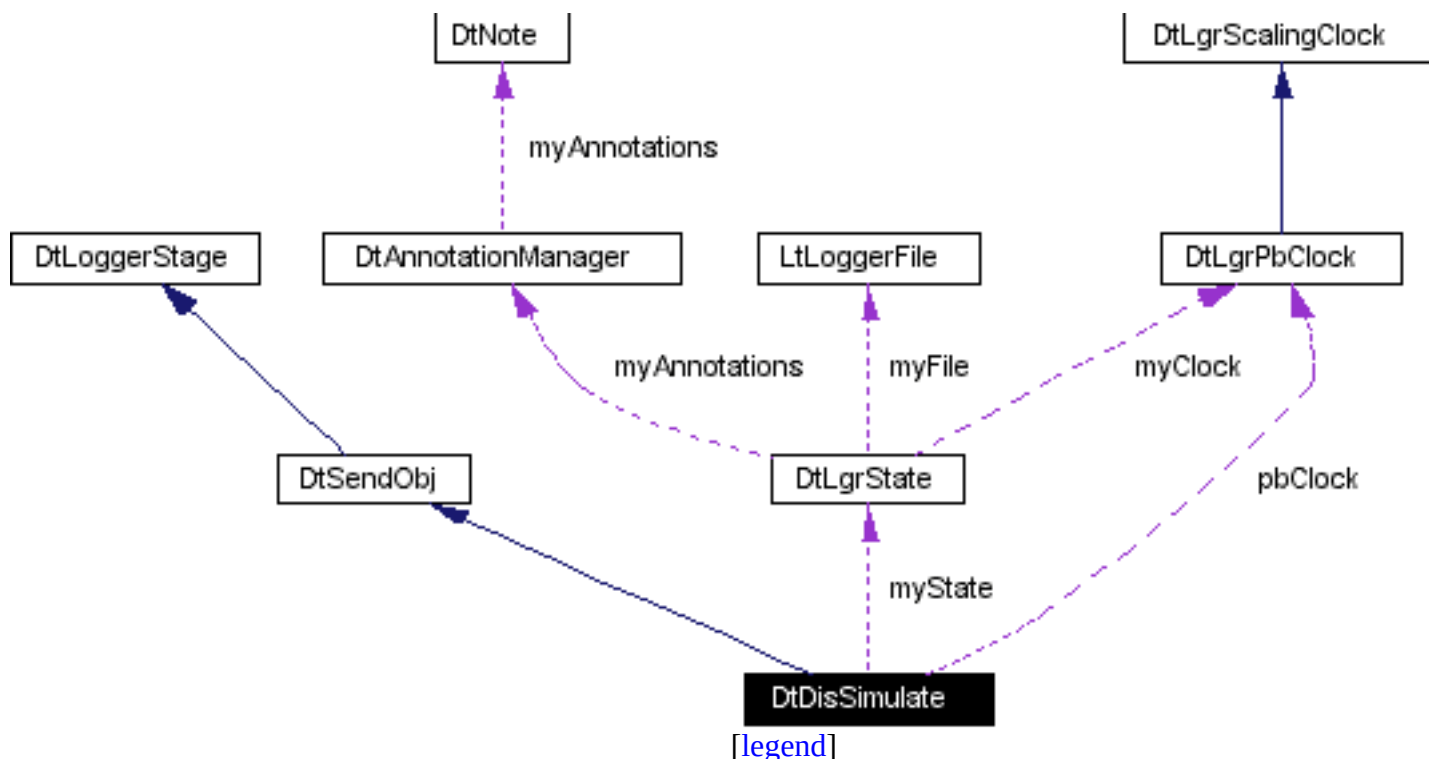
```
#include <disSim.h>
```

Inherits **DtSendObj**.

Inheritance diagram for DtDisSimulate:



Collaboration diagram for DtDisSimulate:



[List of all members.](#)

Public Methods

- **DtDisSimulate** (DtString theName)
- **~DtDisSimulate** ()
- virtual void **initSelf** (**DtLgrState** *theState)
this method is called once after construction to perform some initial initialization. [More...](#)
- virtual void **activate** ()
this method is called when the logger agent has selected to send data to this object. [More...](#)
- virtual void **deactivate** ()
this method is called when the logger agent is no longer going to send data to this object. [More...](#)
- virtual void **processData** (const DtList *packetList)
this method is how the logger will send data to the object. [More...](#)
- virtual void **sendPacket** (const **LtPacketInfo** *packet)
this method is where the send object performs most of its work. [More...](#)
- virtual const char* **printInfo** () const
this method returns a string that contains identifying info about the object. [More...](#)

Static Public Methods

- void* **create** (DtString param)

Protected Methods

- virtual void **send** (**LtPacketInfo** *info)
 - void **procEsPdu** (char *buff, int, DtBoolean sendFlag)
 - void **procAggPdu** (char *buff, int, DtBoolean sendFlag)
 - virtual void **deleteAllObjs** ()
 - virtual void **tick** ()
this method gives the object execution time. [More...](#)
 - virtual void **forceUpdatesOnAll** ()
-

Constructor & Destructor Documentation

DtDisSimulate::DtDisSimulate (DtString *theName*)

DtDisSimulate::~~DtDisSimulate ()

Member Function Documentation

void DtDisSimulate::activate () [virtual]

this method is called when the logger agent has selected to send data to this object.

Reimplemented from [DtSendObj](#).

void * DtDisSimulate::create (DtString *param*) [static]

void DtDisSimulate::deactivate () [virtual]

this method is called when the logger agent is no longer going to send data to this object.

Reimplemented from [DtSendObj](#).

void DtDisSimulate::deleteAllObjs () [protected, virtual]

void DtDisSimulate::forceUpdatesOnAll () [protected, virtual]

```
void DtDisSimulate::initSelf (DtLgrState * theState ) [virtual]
```

this method is called once after construction to perform some initial initialization.

Reimplemented from **DtSendObj**.

```
const char * DtDisSimulate::printInfo () const [virtual]
```

this method returns a string that contains identifying info about the object.

Reimplemented from **DtSendObj**.

```
void DtDisSimulate::procAggPdu (char * buff,  
                                int,  
                                DtBoolean sendFlag ) [protected]
```

```
void DtDisSimulate::procEsPdu (char * buff,  
                               int,  
                               DtBoolean sendFlag ) [protected]
```

```
void DtDisSimulate::processData (const DtList * packetList ) [virtual]
```

this method is how the logger will send data to the object.

Reimplemented from **DtSendObj**.

```
void DtDisSimulate::send (LtPacketInfo * info ) [protected, virtual]
```

```
void DtDisSimulate::sendPacket (const LtPacketInfo * packet ) [inline, virtual]
```

this method is where the send object performs most of its work.

Reimplemented from **DtSendObj**.

```
void DtDisSimulate::tick () [protected, virtual]
```

this method gives the object execution time.

Reimplemented from **DtSendObj**.

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

- [disSim.h](#)

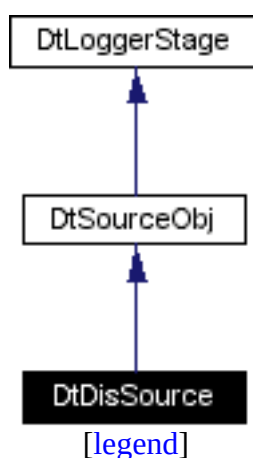
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtDisSource Class Reference

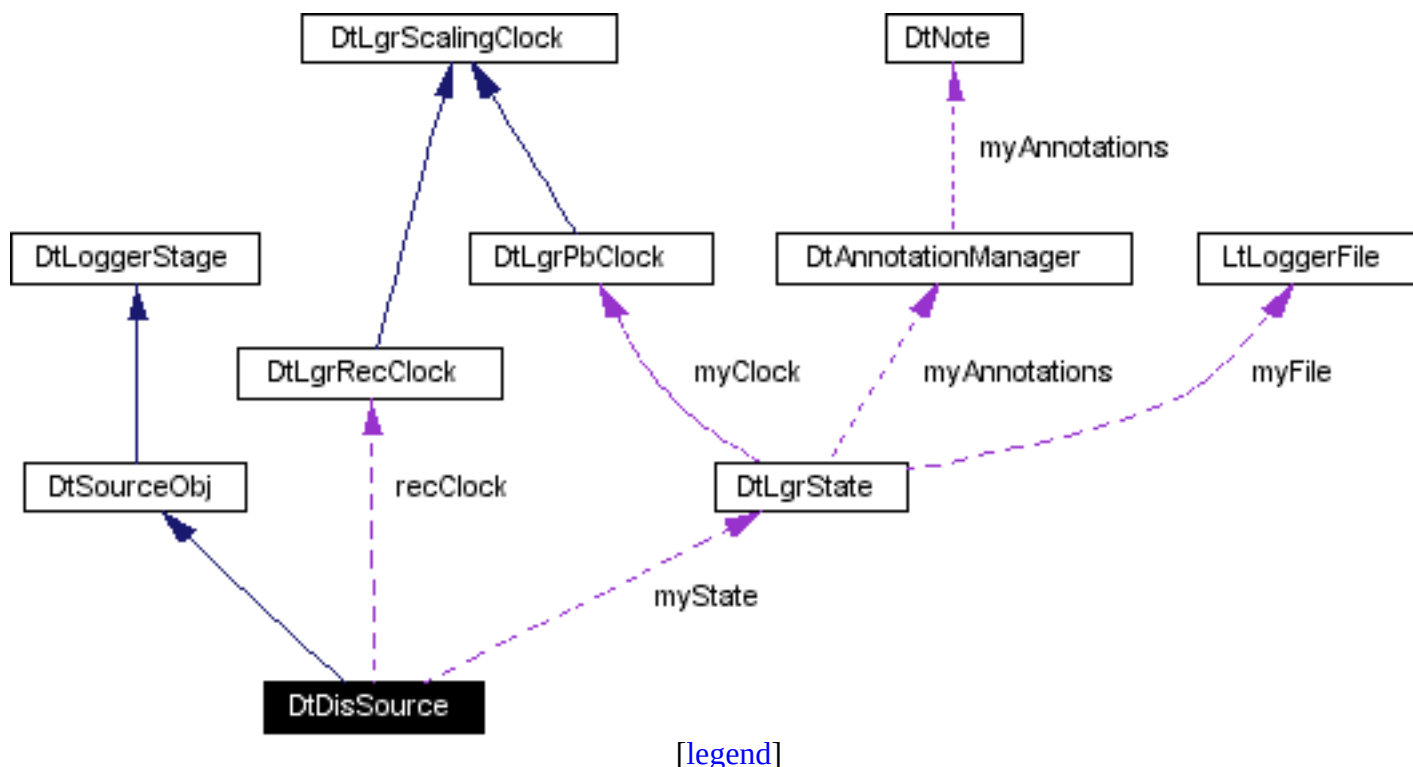
#include <disSource.h>

Inherits **DtSourceObj**.

Inheritance diagram for DtDisSource:



Collaboration diagram for DtDisSource:



[List of all members.](#)

Public Methods

- **DtDisSource** (DtString theName)
- virtual **~DtDisSource** ()
- virtual void **initSelf** (**DtLgrState** *theAIF)
this method is called soon after object creation to do begining initialization. [More...](#)
- virtual DtTime **activate** (DtTime startTime)
the Agent will call this method when it wishes to begin receiving data from this source. [More...](#)
- virtual void **activate** ()
this method is called when the logger agent has selected to send data to this object. [More...](#)
- virtual void **deactivate** ()
The agent will call this method when it no longer wishes to receive data from this source. [More...](#)
- virtual DtTime **generatePackets** (DtTime currentTime, DtList *packetList)
The agent calls this method to retrieve data from the source. the packet list is the current buffer of unprocessed packets from all sources. This Obj should add its packets to the end of the list. Also It should process all packets it has ready. The return value is an indication of when the next packet will be ready. sources such as data files should return the time for the next packet in the file. Real time sources should return negative to indicate the time of the next packet is unknown. [More...](#)
- virtual **LtPacketInfo*** **generatePacket** (DtTime currentTime, DtTime *nextTime)
this method returns one new packet that is ready. if no packet is ready it returns NULL. The time of the next packet is returned through nextTime parameter. [More...](#)
- virtual void **receivePdu** (DtPdu *pdu)
- virtual void **startListening** ()
- virtual const char* **printInfo** () const
this method returns a string that contains identifying info about the object. [More...](#)
- virtual void **reset** ()
this method is called to indicate the object should reset itself to its initial state. Eg a file should return to the beginning. [More...](#)

Static Public Methods

- void **receivePduCb** (DtPdu *pdu, void *usr)
- void* **create** (DtString param)

Protected Attributes

- DtBoolean **myListenFlag**
 - DtList **myPacketList**
 - **DtLgrRecClock*** **recClock**
 - **DtLgrState*** **myState**
 - DtBoolean **myActive**
-

Constructor & Destructor Documentation

```
DtDisSource::DtDisSource (DtString theName )
```

```
DtDisSource::~~DtDisSource () [virtual]
```

Member Function Documentation

```
void DtDisSource::activate () [inline, virtual]
```

this method is called when the logger agent has selected to send data to this object.

Reimplemented from **DtSourceObj**.

```
DtTime DtDisSource::activate (DtTime startTime ) [virtual]
```

the Agent will call this method when it wishes to begin receiving data from this source.

Reimplemented from **DtSourceObj**.

```
void * DtDisSource::create (DtString param ) [static]
```

```
void DtDisSource::deactivate () [virtual]
```

The agent will call this method when it no longer wishes to receive data from this source.

Reimplemented from **DtSourceObj**.

```
LTPacketInfo * DtDisSource::generatePacket (DtTime currentTime,  
                                             DtTime * nextTime) [inline, virtual]
```

this method returns one new packet that is ready. if no packet is ready it returns NULL. The time of the next packet is returned through nextTime parameter.

Reimplemented from **DtSourceObj**.

```
DtTime DtDisSource::generatePackets (DtTime currentTime,  
                                     DtList * packetList) [virtual]
```

The agent calls this method to retrieve data from the source. the packet list is the current buffer of unprocessed packets from all sources. This Obj should add its packets to the end of the list. Also It should process all packets it has ready. The return value is an indication of when the next packet will be ready. sources such as data files should return the time for the next packet in the file. Real time sources should return negative to indicate the time of the next packet is unknown.

Reimplemented from **DtSourceObj**.

```
void DtDisSource::initSelf (DtLgrState * theAIF) [virtual]
```

this method is called soon after object creation to do beginning initialization.

Reimplemented from **DtSourceObj**.

```
const char * DtDisSource::printInfo () const [virtual]
```

this method returns a string that contains identifying info about the object.

Reimplemented from **DtSourceObj**.

```
void DtDisSource::receivePdu (DtPdu * pdu ) [virtual]
```

```
void DtDisSource::receivePduCb (DtPdu * pdu,  
                                void * usr ) [static]
```

```
void DtDisSource::reset () [virtual]
```

this method is called to indicate the object should reset itself to its initial state. Eg a file should return to the beginning.

Reimplemented from **DtSourceObj**.

```
void DtDisSource::startListening () [virtual]
```

Member Data Documentation

```
DtBoolean DtDisSource::myActive [protected]
```

```
DtBoolean DtDisSource::myListenFlag [protected]
```

```
DtList DtDisSource::myPacketList [protected]
```

```
DtLgrState * DtDisSource::myState [protected]
```

DtLgrRecClock * DtDisSource::recClock [protected]

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

- **[disSource.h](#)**

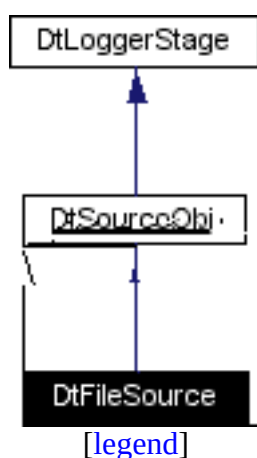
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtFileSource Class Reference

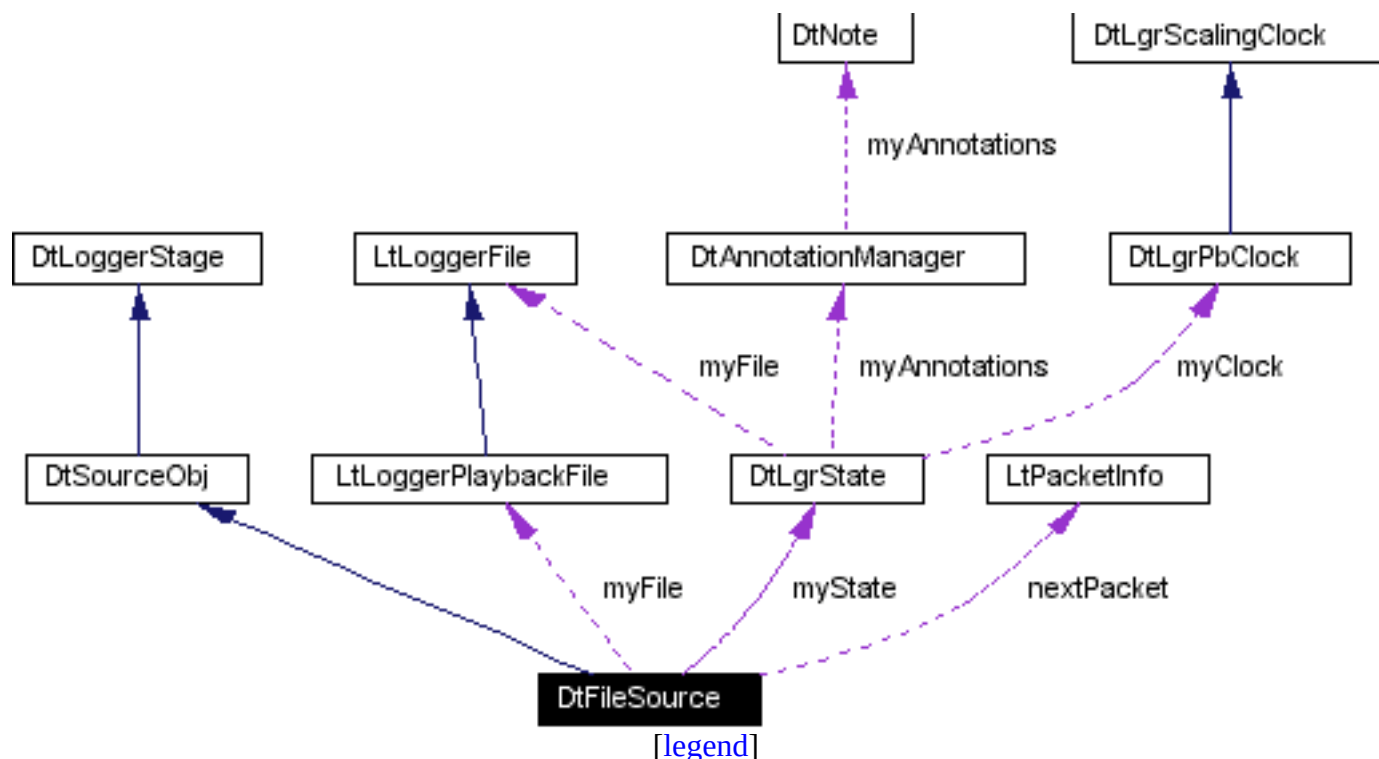
#include <fileSource.h>

Inherits **DtSourceObj**.

Inheritance diagram for DtFileSource:



Collaboration diagram for DtFileSource:



[List of all members.](#)

Public Methods

- **DtFileSource** (DtString theName)
- virtual **~DtFileSource** ()
- virtual void **initSelf** (**DtLgrState** *theState)
this method is called soon after object creation to do begining initialization. [More...](#)
- virtual DtTime **activate** (DtTime startTime)
the Agent will call this method when it wishes to receive data from this source returns when the first packet will be available, negative if unknown. [More...](#)
- virtual void **activate** ()
this method is called when the logger agent has selected to send data to this object. [More...](#)
- virtual void **deactivate** ()
The agent will call this method when it no longer wishes to receive data from this source. [More...](#)
- virtual DtTime **generatePackets** (DtTime currentTime, DtList *packetList)
The agent calls this method to retrieve data from the source. the packet list is the current buffer of unprocessed packets from all sources. This Obj should add its packets to the end of the list. Also It should process all packets it has ready. The return value is an indication of when the next packet will be ready. sources such as data files should return the time for the next packet in the file. Real time sources should return negative to indicate the time of the next packet is unknown. [More...](#)
- virtual **LtPacketInfo*** **generatePacket** (DtTime currentTime, DtTime *nextTime)
this method returns one new packet that is ready. if no packet is ready it returns NULL. The time of the next packet is returned through nextTime parameter. [More...](#)
- virtual const char* **printInfo** () const
this method returns a string that contains identifying info about the object. [More...](#)
- virtual void **reset** ()
this method is called to indicate the object should reset itself to its initial state. Eg a file should return to the beginning. [More...](#)
- virtual void **tick** ()
this method is called to give this object execution time. [More...](#)

Static Public Methods

- void* **create** (DtString param)
factory method. [More...](#)
-

Constructor & Destructor Documentation

DtFileSource::DtFileSource (DtString *theName*)

DtFileSource::~~DtFileSource () [virtual]

Member Function Documentation

void DtFileSource::activate () [inline, virtual]

this method is called when the logger agent has selected to send data to this object.

Reimplemented from **DtSourceObj**.

DtTime DtFileSource::activate (DtTime *startTime*) [virtual]

the Agent will call this method when it wishes to receive data from this source returns when the first packet will be available, negative if unknown.

Reimplemented from **DtSourceObj**.

void * DtFileSource::create (DtString *param*) [static]

factory method.

void DtFileSource::deactivate () [virtual]

The agent will call this method when it no longer wishes to receive data from this source.

Reimplemented from **DtSourceObj**.

```
LtPacketInfo * DtFileSource::generatePacket (DtTime currentTime,  
                                              DtTime * nextTime) [inline, virtual]
```

this method returns one new packet that is ready. if no packet is ready it returns NULL. The time of the next packet is returned through *nextTime* parameter.

Reimplemented from **DtSourceObj**.

```
DtTime DtFileSource::generatePackets (DtTime currentTime,  
                                       DtList * packetList) [virtual]
```

The agent calls this method to retrieve data from the source. the packet list is the current buffer of unprocessed packets from all sources. This Obj should add its packets to the end of the list. Also It should process all packets it has ready. The return value is an indication of when the next packet will be ready. sources such as data files should return the time for the next packet in the file. Real time sources should return negative to indicate the time of the next packet is unknown.

Reimplemented from **DtSourceObj**.

```
void DtFileSource::initSelf (DtLgrState * theState) [virtual]
```

this method is called soon after object creation to do begining initialization.

Reimplemented from **DtSourceObj**.

```
const char * DtFileSource::printInfo () const [virtual]
```

this method returns a string that contains identifying info about the object.

Reimplemented from **DtSourceObj**.

void DtFileSource::reset () [virtual]

this method is called to indicate the object should reset itself to its initial state. Eg a file should return to the beginning.

Reimplemented from **DtSourceObj**.

void DtFileSource::tick () [virtual]

this method is called to give this object execution time.

Reimplemented from **DtSourceObj**.

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

- **[fileSource.h](#)**

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

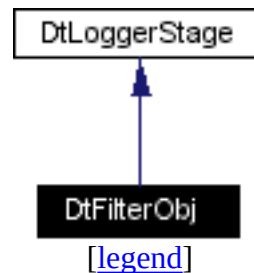
DtFilterObj Class Reference

this is an abstract class that defines a filter for the logger agent to manage data. examples can be ramToools interface. [More...](#)

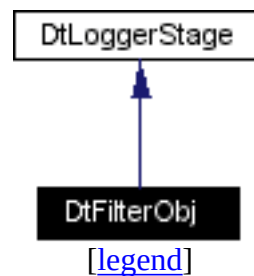
```
#include <filterObj.h>
```

Inherits **DtLoggerStage**.

Inheritance diagram for DtFilterObj:



Collaboration diagram for DtFilterObj:



[List of all members.](#)

Public Methods

- **DtFilterObj** (const char *theName)
- virtual **~DtFilterObj** ()
- virtual void **initSelf** (**DtLgrState** *theState)=0
this method is called once after construction to perform some initial initialization. [More...](#)
- virtual void **activate** ()=0

this method is called when the logger agent has selected to send data to this object. [More...](#)

- virtual void **deactivate** ()=0

this method is called when the logger agent is no longer going to send data to this object. [More...](#)

- virtual void **processData** (DtList *packetList)

the logger send the data as a list to the filter object in this method. The default method will send each packet to filterPacket and if Filter packet returns DtTrue the packet will be removed from the list. [More...](#)

- virtual DtBoolean **filterPacket** (LtPacketInfo *packet)=0

this method is where the filtering is performed. A return value of DtTrue will cause the packet to be removed by processData. [More...](#)

- virtual void **tick** ()

this method gives the object execution time. [More...](#)

- virtual DtBoolean **isOk** () const

- virtual const char* **getName** () const

- virtual const char* **printInfo** () const=0

this method returns a string that supplies identifying info about the object. [More...](#)

Protected Attributes

- DtBoolean **myStatus**
-

Detailed Description

this is an abstract class that defines a filter for the logger agent to manage data. examples can be ramToools interface.

Constructor & Destructor Documentation

```
DtFilterObj::DtFilterObj (const char * theName )
```

DtFilterObj::~~DtFilterObj () [virtual]

Member Function Documentation

void DtFilterObj::activate () [pure virtual]

this method is called when the logger agent has selected to send data to this object.

Reimplemented from [DtLoggerStage](#).

void DtFilterObj::deactivate () [pure virtual]

this method is called when the logger agent is no longer going to send data to this object.

Reimplemented from [DtLoggerStage](#).

DtBoolean DtFilterObj::filterPacket ([LtPacketInfo](#) * *packet*) [pure virtual]

this method is where the filtering is performed. A return value of DtTrue will cause the packet to be removed by processData.

const char * DtFilterObj::getName () const [inline, virtual]

Reimplemented from [DtLoggerStage](#).

void DtFilterObj::initSelf ([DtLgrState](#) * *theState*) [pure virtual]

this method is called once after construction to perform some initial initialization.

Reimplemented from [DtLoggerStage](#).

DtBoolean DtFilterObj::isOk () const [inline, virtual]

Reimplemented from [DtLoggerStage](#).

```
const char * DtFilterObj::printInfo () const [pure virtual]
```

this method returns a string that supplies identifying info about the object.

Reimplemented from [DtLoggerStage](#).

```
void DtFilterObj::processData (DtList * packetList ) [virtual]
```

the logger send the data as a list to the filter object in this method. The default method will send each packet to filterPacket and if Filter packet returns DtTrue the packet will be removed from the list.

```
void DtFilterObj::tick () [inline, virtual]
```

this method gives the object execution time.

Reimplemented from [DtLoggerStage](#).

Member Data Documentation

```
DtBoolean DtFilterObj::myStatus [protected]
```

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

- [filterObj.h](#)

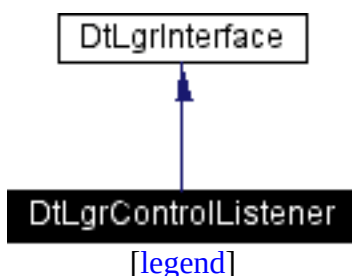
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtLgrControlListener Class Reference

```
#include <lgrCtrlListener.h>
```

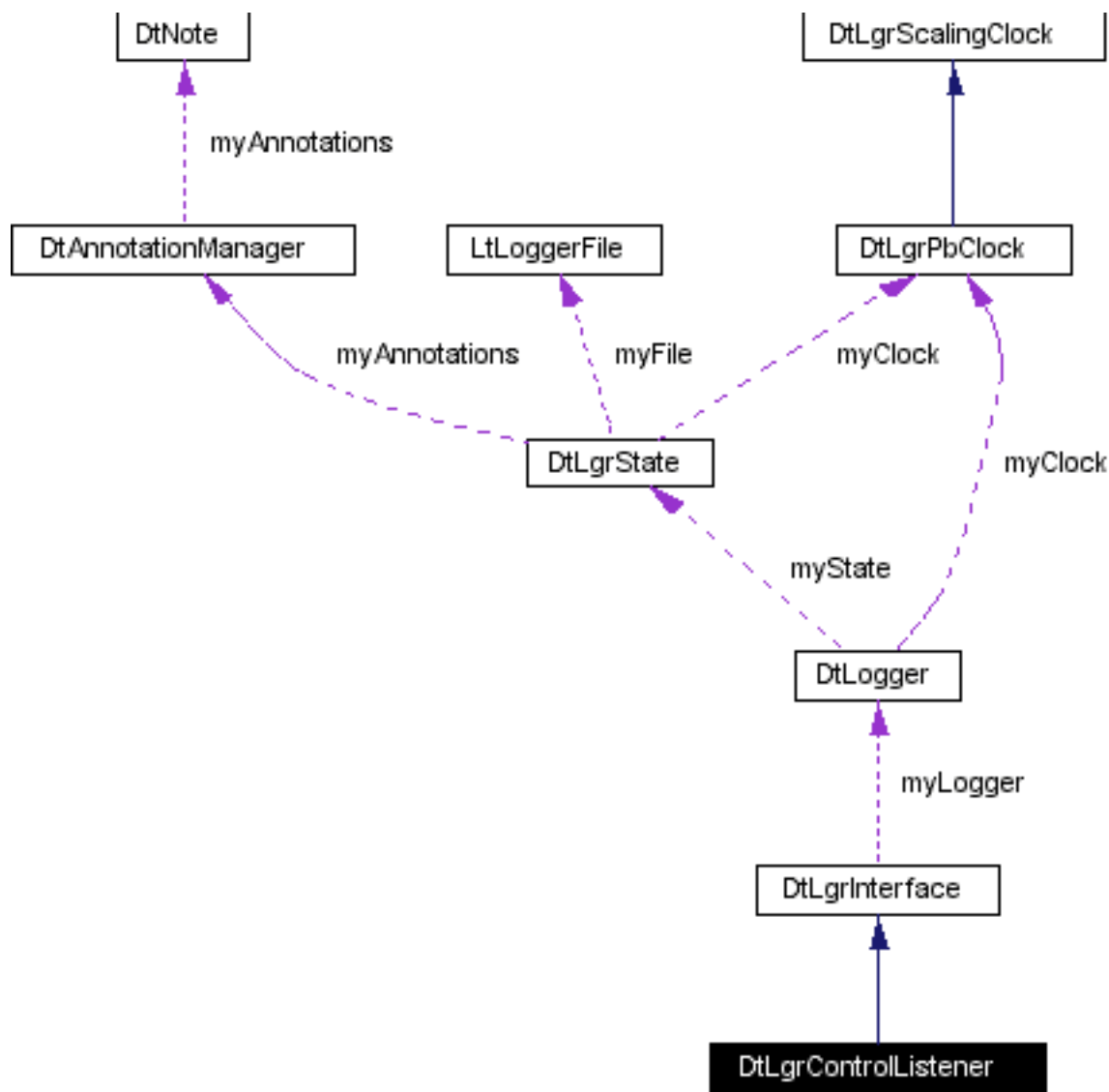
Inherits **DtLgrInterface**.

Inheritance diagram for DtLgrControlListener:



[\[legend\]](#)

Collaboration diagram for DtLgrControlListener:



[[legend](#)]

[List of all members.](#)

Public Methods

- **DtLgrControlListener** (**DtLogger** *logger, DtString id, DtBoolean processAll=DtTrue)
- virtual ~**DtLgrControlListener** ()
- virtual void **setAllowFileDeletion** (DtBoolean yesNo)

Set/Get whether the listener will delete files upon receipt of a DtLgrDelFilePdu/Interaction. The logger turns this on based on whether allowRemoteFileDelete is set in logger.mtl AND the -D option is specified on the cmd line.

[More...](#)

- virtual DtBoolean **allowingFileDeletion** () const
- virtual void **setProcessAll** (DtBoolean yesNo)

Set/Get whether we process all LgrCtrlPdu's or just our own. [More...](#)

- virtual DtBoolean **processAll** () const

- virtual void **setId** (const DtString &id)
Set/Get our ID. [More...](#)
- virtual const DtString& **id** () const

Static Public Methods

- void **lgrControlCb** (DtInteraction *lcInter, void *theAction)

Public Attributes

- DtBoolean **myProcessAll**
- DtBoolean **myAllowFileDelete**
- DtString **myId**

Static Protected Methods

- void **processLgrControl** (void *usr)

Constructor & Destructor Documentation

```
DtLgrControlListener::DtLgrControlListener (DtLogger * logger,  
                                             DtString id,  
                                             DtBoolean processAll = DtTrue )
```

```
DtLgrControlListener::~DtLgrControlListener () [virtual]
```

Member Function Documentation

```
DtBoolean DtLgrControlListener::allowingFileDeletion () const [virtual]
```

```
const DtString & DtLgrControlListener::id () const [virtual]
```

```
void DtLgrControlListener::lgrControlCb (DtInteraction * lcInter,  
void * theAction) [static]
```

```
DtBoolean DtLgrControlListener::processAll () const [virtual]
```

```
void DtLgrControlListener::processLgrControl (void * usr) [static, protected]
```

```
void DtLgrControlListener::setAllowFileDeletion (DtBoolean yesNo) [virtual]
```

Set/Get whether the listener will delete files upon receipt of a DtLgrDelFilePdu/Interaction. The logger turns this on based on whether allowRemoteFileDelete is set in logger.mtl AND the -D option is specified on the cmd line.

```
void DtLgrControlListener::setId (const DtString & id) [virtual]
```

Set/Get our ID.

```
void DtLgrControlListener::setProcessAll (DtBoolean yesNo) [virtual]
```

Set/Get whether we process all LgrCtrlPdu's or just our own.

Member Data Documentation

```
DtBoolean DtLgrControlListener::myAllowFileDelete
```

DtString DtLgrControlListener::myId

DtBoolean DtLgrControlListener::myProcessAll

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

- [**lgrCtrlListener.h**](#)

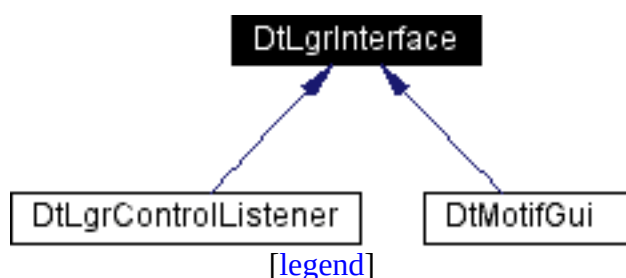
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtLgrInterface Class Reference

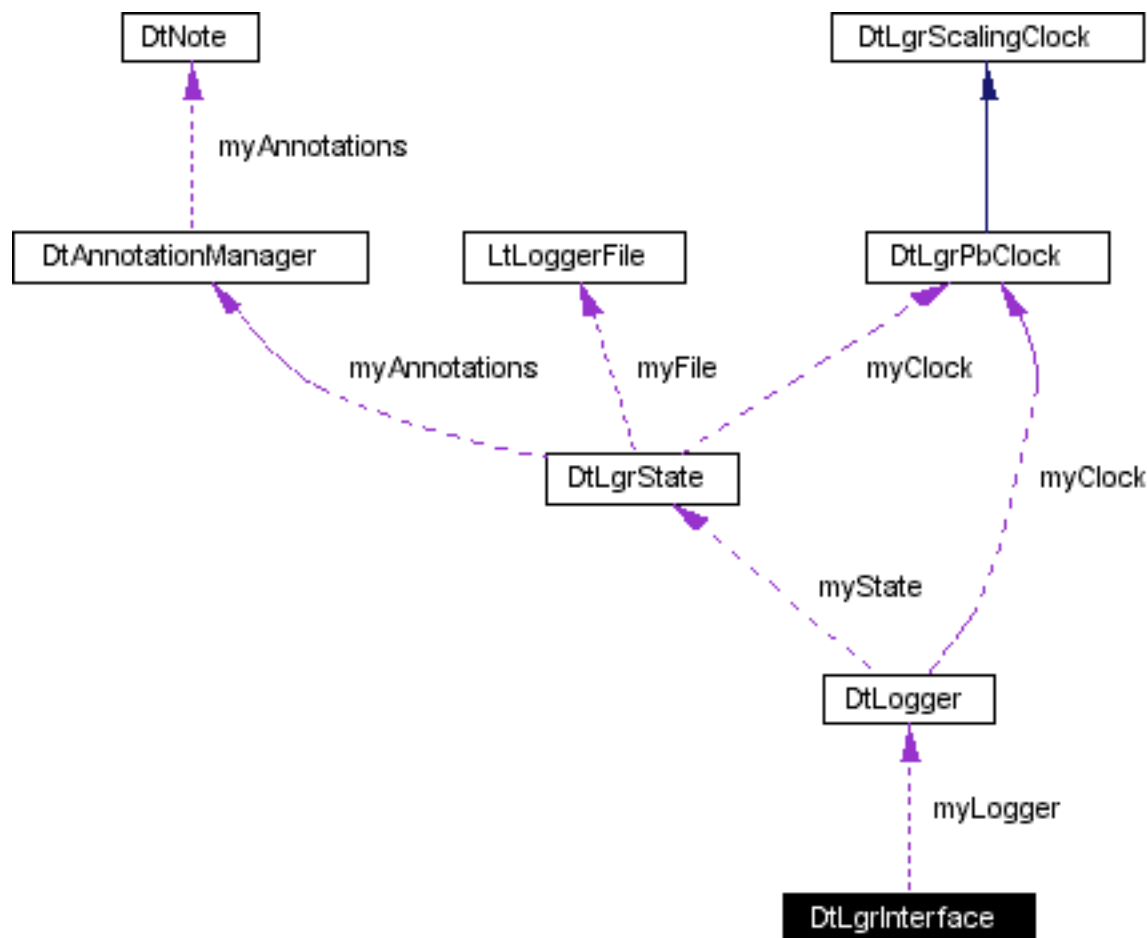
```
#include <lgrInterface.h>
```

Inherited by **DtLgrControlListener**, and **DtMotifGui**.

Inheritance diagram for DtLgrInterface:



Collaboration diagram for DtLgrInterface:



[List of all members.](#)

Public Methods

- **DtLgrInterface** (**DtLogger** *logger)
- virtual ~**DtLgrInterface** ()
- virtual DtBoolean **timeToQuit** ()
- virtual void **setTimeToQuit** (DtBoolean yesNo)
- virtual void **tick** ()

Public Attributes

- **DtLogger*** **myLogger**

Constructor & Destructor Documentation

```
DtLgrInterface::DtLgrInterface (DtLogger * logger ) [inline]
```

```
DtLgrInterface::~~DtLgrInterface () [inline, virtual]
```

Member Function Documentation

```
void DtLgrInterface::setTimeToQuit (DtBoolean yesNo ) [inline, virtual]
```

```
void DtLgrInterface::tick () [inline, virtual]
```

Reimplemented in **DtMotifGui**.

DtBoolean DtLgrInterface::timeToQuit () [inline, virtual]

Member Data Documentation

[DtLogger](#) * DtLgrInterface::myLogger

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

- [lgrInterface.h](#)

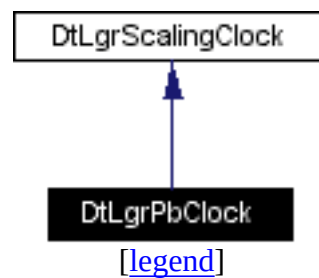
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtLgrPbClock Class Reference

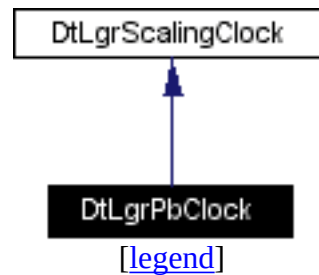
```
#include <lgrClock.h>
```

Inherits [DtLgrScalingClock](#).

Inheritance diagram for DtLgrPbClock:



Collaboration diagram for DtLgrPbClock:



[List of all members.](#)

Public Methods

- [DtLgrPbClock](#) (DtClock *c)
- virtual [~DtLgrPbClock](#) ()
- DtTime [getConstScaledTime](#) ()
- void [setConstScaledTime](#) (DtTime t)
- DtTime [getConstraint](#) ()
- void [setConstraint](#) (DtTime c)
- void [setTimeScale](#) (double ts)

Constructor & Destructor Documentation

DtLgrPbClock::DtLgrPbClock (DtClock * *c*)

DtLgrPbClock::~~DtLgrPbClock () [virtual]

Member Function Documentation

DtTime DtLgrPbClock::getConstScaledTime ()

DtTime DtLgrPbClock::getConstraint ()

void DtLgrPbClock::setConstScaledTime (DtTime *t*)

void DtLgrPbClock::setConstraint (DtTime *c*)

void DtLgrPbClock::setTimeScale (double *ts*) [virtual]

Reimplemented from **DtLgrScalingClock**.

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

- **[lgrClock.h](#)**

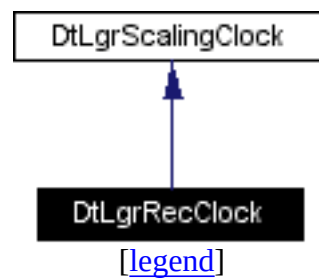
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtLgrRecClock Class Reference

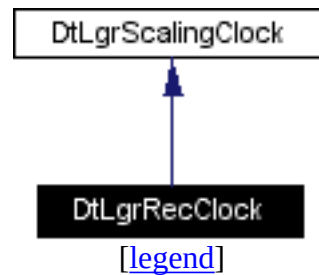
`#include <lgrClock.h>`

Inherits **DtLgrScalingClock**.

Inheritance diagram for DtLgrRecClock:



Collaboration diagram for DtLgrRecClock:



[List of all members.](#)

Public Methods

- **DtLgrRecClock** (DtClock *c)
- virtual **~DtLgrRecClock** ()
- void **setTimeScale** (double ts)

Constructor & Destructor Documentation

DtLgrRecClock::DtLgrRecClock (DtClock * *c*)

DtLgrRecClock::~~DtLgrRecClock () [virtual]

Member Function Documentation

void DtLgrRecClock::setTimeScale (double *ts*) [virtual]

Reimplemented from **[DtLgrScalingClock](#)**.

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

- **[lgrClock.h](#)**

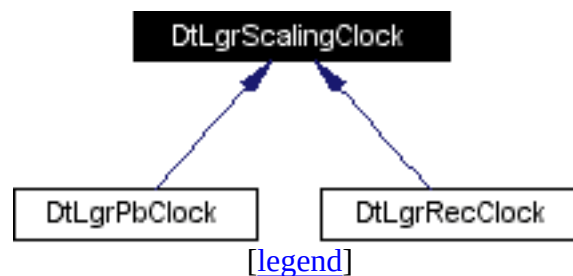
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtLgrScalingClock Class Reference

```
#include <lgrClock.h>
```

Inherited by **DtLgrPbClock**, and **DtLgrRecClock**.

Inheritance diagram for DtLgrScalingClock:



[List of all members.](#)

Public Methods

- **DtLgrScalingClock** (DtClock *c)
Constructor. [More...](#)
- virtual **~DtLgrScalingClock** ()
Destructor. [More...](#)
- DtTime **getIdealScaledTime** ()
Set/Get. [More...](#)
- void **setIdealScaledTime** (DtTime t)
- DtTime **idealScaledToReal** (DtTime scaledTime)
- DtTime **realToIdealScaled** (DtTime realTime)
- double **getTimeScale** ()
- DtTime **getRealTime** ()

Protected Methods

- virtual void **setTimeScale** (double ts)=0

Protected Attributes

- double **timeScale**
 - DtTime **idealOffset**
 - DtClock* **myClock**
-

Constructor & Destructor Documentation

DtLgrScalingClock::DtLgrScalingClock (DtClock * c)

Constructor.

DtLgrScalingClock::~~DtLgrScalingClock () [virtual]

Destructor.

Member Function Documentation

DtTime DtLgrScalingClock::getIdealScaledTime ()

Set/Get.

double DtLgrScalingClock::getRealTime () [inline]

double DtLgrScalingClock::getTimeScale ()

DtTime DtLgrScalingClock::idealScaledToReal (DtTime *scaledTime*)

DtTime DtLgrScalingClock::realToIdealScaled (DtTime *realTime*)

void DtLgrScalingClock::setIdealScaledTime (DtTime *t*)

void DtLgrScalingClock::setTimeScale (double *ts*) [protected, pure virtual]

Reimplemented in [DtLgrPbClock](#), and [DtLgrRecClock](#).

Member Data Documentation

DtTime DtLgrScalingClock::idealOffset [protected]

DtClock * DtLgrScalingClock::myClock [protected]

double DtLgrScalingClock::timeScale [protected]

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

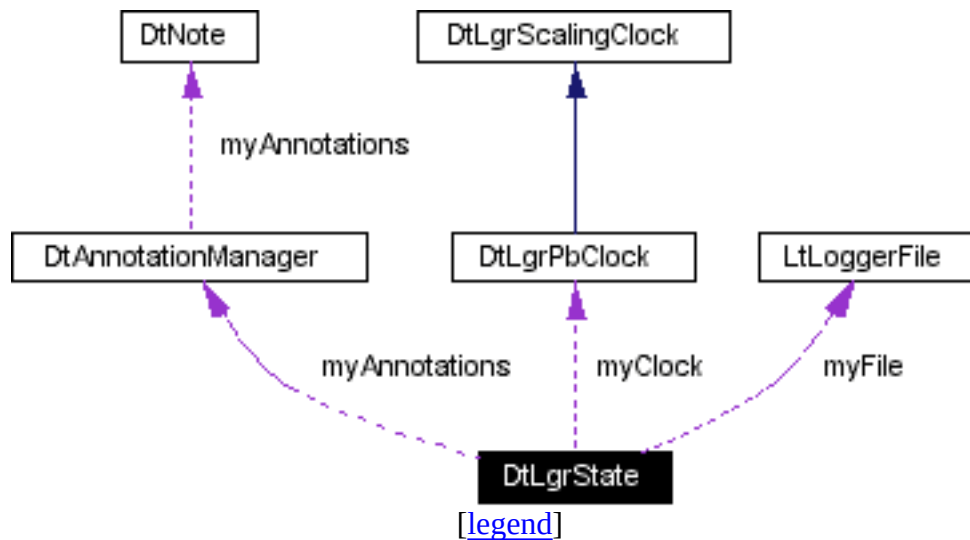
- [lgrClock.h](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtLgrState Class Reference

```
#include <lgrState.h>
```

Collaboration diagram for DtLgrState:



[List of all members.](#)

Public Methods

- **DtLgrState** (DtExerciseConn *exConn, **DtLgrPbClock** *clock)
- virtual **~DtLgrState** ()
- virtual void **clearFlags** ()
Clear source file control variables. [More...](#)
- virtual DtBoolean **skipToEnd** () const
Check source file control variables. [More...](#)
- virtual DtBoolean **skipToStart** () const
- virtual int **timeJumped** () const
- virtual void **setSkipToEnd** ()
Set source file control variables. [More...](#)
- virtual void **setSkipToStart** ()

- virtual void **setTimeJumped** (int state=2)
- virtual DtLgrFilePosition **filePosition** () const
Get/Set source file position. [More...](#)
- virtual void **setFilePosition** (DtLgrFilePosition pos)
- virtual **LtLoggerFile*** **file** () const
Get/Set source/send file. [More...](#)
- virtual void **setFile** (**LtLoggerFile** *f)
- virtual DtLgrMode **mode** () const
Get/Set modes. [More...](#)
- virtual void **setMode** (DtLgrMode m)
- virtual DtLgrMode **eofAction** () const
Get/Set the EOF action. [More...](#)
- virtual void **setEofAction** (DtLgrMode e)
- virtual DtLgrMode **playAction** () const
Get/Set the Play action. [More...](#)
- virtual void **setPlayAction** (DtLgrMode m)
- virtual DtLgrMode **recordAction** () const
Get/Set the Record Action. [More...](#)
- virtual void **setRecordAction** (DtLgrMode m)
- virtual DtBoolean **dataScaling** () const
Get/Set data scaling. [More...](#)
- virtual void **setDataScaling** (DtBoolean onOff)
- virtual int **packetLimit** () const
Get/Set source limit on number of packets to generate If the limit is set to -1, no cap will be put on the number of packets generated; The default limit is DEFAULT_PACKET_LIMIT. [More...](#)
- virtual void **setPacketLimit** (int val)
- virtual void **postToClipboard** (DtString keyname, void *data)
Clipboard functions. [More...](#)
- virtual void* **fetchFromClipboard** (DtString keyname)
- virtual void* **removeFromClipboard** (DtString keyname)
- virtual **DtAnnotationManager*** **annotations** ()
accessors. [More...](#)
- virtual **DtLgrPbClock*** **clock** () const
- virtual DtHashlist* **stateReps** ()

- virtual DtExerciseConn* **exerciseConn** () const
- virtual void **setPlaybackFileCreator** (DtLoggerPlaybackConstructMethod m)
- virtual void **setRecordFileCreator** (DtLoggerRecordConstructMethod m)
- virtual **DtLoggerPlaybackConstructMethod** playbackFileCreator ()
- virtual **DtLoggerRecordConstructMethod** recordFileCreator ()
- virtual double **timeout** () const
Set/Get timeout. [More...](#)
- virtual void **setTimeout** (double val)
- virtual double **heartBeat** () const
Set/Get heartbeat. [More...](#)
- virtual void **setHeartBeat** (double val)

Protected Attributes

- DtExerciseConn* **myExConn**
*Ptrs to members in **DtLogger**. [More...](#)*
- **DtLgrPbClock*** **myClock**
- **LtLoggerFile*** **myFile**
Ptr to logger file. [More...](#)
- DtHashlist **myStateRep**
Used for data scaling. [More...](#)
- int **myTimeJumped**
control variables. [More...](#)
- DtBoolean **mySkipToEnd**
- DtBoolean **mySkipToStart**
- DtLgrFilePosition **myFilePosition**
state variables. [More...](#)
- DtTime **myFileLength**
- DtBoolean **myDataScaling**
- DtLgrMode **myMode**
- DtLgrMode **myPlayAction**
- DtLgrMode **myRecordAction**
- DtLgrMode **myEofAction**
- DtHashlist **myClipboard**

Common repository for objects to post data which other objects might need. [More...](#)

- [DtLoggerPlaybackConstructMethod](#) [myPlaybackConstructMethod](#)
- [DtLoggerRecordConstructMethod](#) [myRecordConstructMethod](#)
- [DtAnnotationManager](#)* [myAnnotations](#)
- int [myPacketLimit](#)

Cap on the number of packets a source object generates It's up to the source object to implement this limit.
[More...](#)

- double [myTimeout](#)
 - double [myHeartbeat](#)
-

Constructor & Destructor Documentation

```
DtLgrState::DtLgrState (DtExerciseConn * exConn,  
                        DtLgrPbClock * clock )
```

```
DtLgrState::~DtLgrState () [virtual]
```

Member Function Documentation

```
DtAnnotationManager * DtLgrState::annotations () [virtual]
```

accessors.

```
void DtLgrState::clearFlags () [virtual]
```

Clear source file control variables.

```
DtLgrPbClock * DtLgrState::clock () const [virtual]
```

DtBoolean DtLgrState::dataScaling () const [virtual]

Get/Set data scaling.

DtLgrMode DtLgrState::eofAction () const [virtual]

Get/Set the EOF action.

DtExerciseConn * DtLgrState::exerciseConn () const [virtual]

void * DtLgrState::fetchFromClipboard (DtString *keyname*) [virtual]

LtLoggerFile * DtLgrState::file () const [virtual]

Get/Set source/send file.

DtLgrFilePosition DtLgrState::filePosition () const [virtual]

Get/Set source file position.

double DtLgrState::heartBeat () const [virtual]

Set/Get heartbeat.

DtLgrMode DtLgrState::mode () const [virtual]

Get/Set modes.

int DtLgrState::packetLimit () const [virtual]

Get/Set source limit on number of packets to generate If the limit is set to -1, no cap will be put on the number of packets generated; The default limit is DEFAULT_PACKET_LIMIT.

DtLgrMode DtLgrState::playAction () const [virtual]

Get/Set the Play action.

DtLoggerPlaybackConstructMethod DtLgrState::playbackFileCreator () [virtual]

**void DtLgrState::postToClipboard (DtString *keyname*,
void * *data*) [virtual]**

Clipboard functions.

DtLgrMode DtLgrState::recordAction () const [virtual]

Get/Set the Record Action.

DtLoggerRecordConstructMethod DtLgrState::recordFileCreator () [virtual]

void * DtLgrState::removeFromClipboard (DtString *keyname*) [virtual]

void DtLgrState::setDataScaling (DtBoolean *onOff*) [virtual]

void DtLgrState::setEofAction (DtLgrMode *e*) [virtual]

```
void DtLgrState::setFile (DtLoggerFile * f) [virtual]
```

```
void DtLgrState::setFilePosition (DtLgrFilePosition pos) [virtual]
```

```
void DtLgrState::setHeartBeat (double val) [virtual]
```

```
void DtLgrState::setMode (DtLgrMode m) [virtual]
```

```
void DtLgrState::setPacketLimit (int val) [virtual]
```

```
void DtLgrState::setPlayAction (DtLgrMode m) [virtual]
```

```
void DtLgrState::setPlaybackFileCreator (DtLoggerPlaybackConstructMethod m) [virtual]
```

```
void DtLgrState::setRecordAction (DtLgrMode m) [virtual]
```

```
void DtLgrState::setRecordFileCreator (DtLoggerRecordConstructMethod m) [virtual]
```

```
void DtLgrState::setSkipToEnd () [virtual]
```

Set source file control variables.

```
void DtLgrState::setSkipToStart () [virtual]
```

```
void DtLgrState::setTimeJumped (int state = 2 ) [virtual]
```

```
void DtLgrState::setTimeout (double val ) [virtual]
```

```
DtBoolean DtLgrState::skipToEnd () const [virtual]
```

Check source file control variables.

```
DtBoolean DtLgrState::skipToStart () const [virtual]
```

```
DtHashlist * DtLgrState::stateReps () [virtual]
```

```
int DtLgrState::timeJumped () const [virtual]
```

```
double DtLgrState::timeout () const [virtual]
```

Set/Get timeout.

Member Data Documentation

```
DtAnnotationManager * DtLgrState::myAnnotations [protected]
```

```
DtHashlist DtLgrState::myClipboard [protected]
```

Common repository for objects to post data which other objects might need.

DtLgrPbClock * DtLgrState::myClock [protected]

DtBoolean DtLgrState::myDataScaling [protected]

DtLgrMode DtLgrState::myEofAction [protected]

DtExerciseConn * DtLgrState::myExConn [protected]

Ptrs to members in **DtLogger**.

LtLoggerFile * DtLgrState::myFile [protected]

Ptr to logger file.

DtTime DtLgrState::myFileLength [protected]

DtLgrFilePosition DtLgrState::myFilePosition [protected]

state variables.

double DtLgrState::myHeartbeat [protected]

DtLgrMode DtLgrState::myMode [protected]

int DtLgrState::myPacketLimit [protected]

Cap on the number of packets a source object generates It's up to the source object to implement this limit.

DtLgrMode DtLgrState::myPlayAction [protected]

[DtLoggerPlaybackConstructMethod](#) DtLgrState::myPlaybackConstructMethod [protected]

DtLgrMode DtLgrState::myRecordAction [protected]

[DtLoggerRecordConstructMethod](#) DtLgrState::myRecordConstructMethod [protected]

DtBoolean DtLgrState::mySkipToEnd [protected]

DtBoolean DtLgrState::mySkipToStart [protected]

DtHashlist DtLgrState::myStateRep [protected]

Used for data scaling.

int DtLgrState::myTimeJumped [protected]

control variables.

double DtLgrState::myTimeout [protected]

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

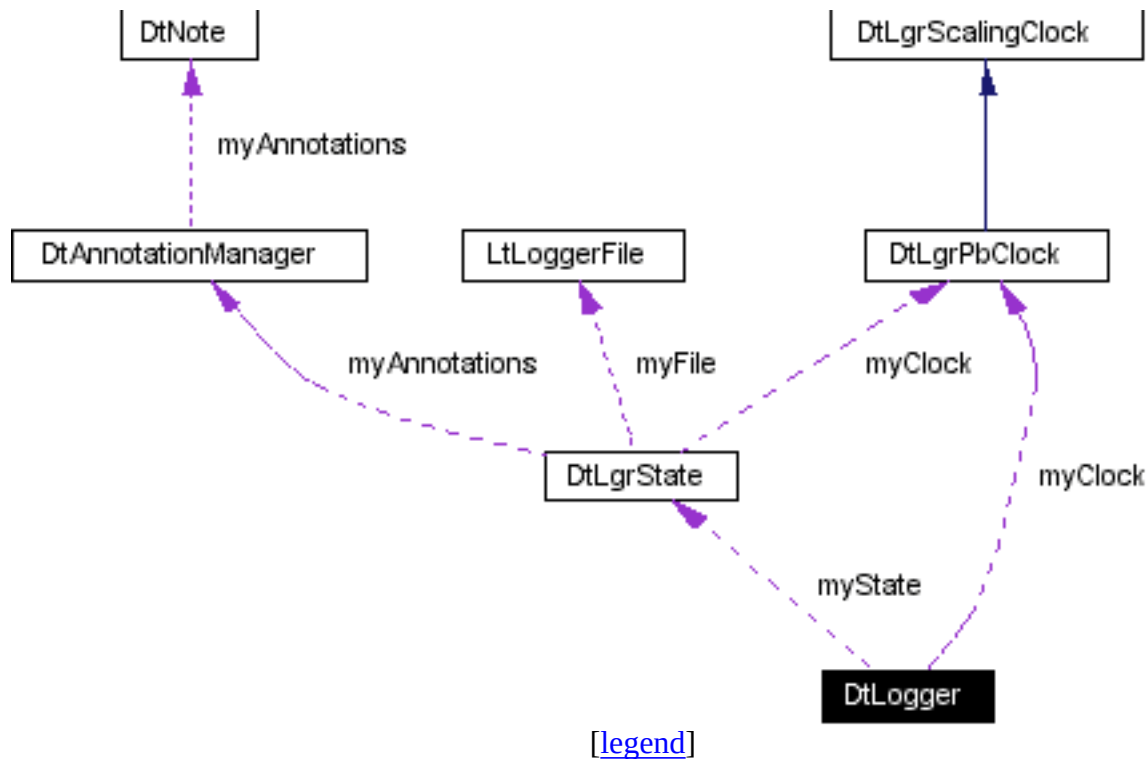
- **[lgrState.h](#)**

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtLogger Class Reference

```
#include <logger.h>
```

Collaboration diagram for DtLogger:



[List of all members.](#)

Public Methods

- **DtLogger** (DtExerciseConn *exConn)
Constructor. [More...](#)
- **DtLogger** (int port, int exercise_id, int site_id, int application_num)
Constructors. [More...](#)
- **DtLogger** (int port, DtInetAddr destAddr, int exercise_id, int site_id, int application_num)
- **DtLogger** (DtSocket *skt, int exercise_id, int site_id, int application_num)
- virtual **~DtLogger** ()

Destructor. [More...](#)

- virtual DtTime **tick** ()
Updates pipeline elements, generates packets, and process packets. Returns the arrival time of the next packet. If the arrival time is unknown, returns -1. [More...](#)
- virtual DtExerciseConn* **exerciseConn** () const
Get exercise conn. [More...](#)
- virtual DtLgrMode **mode** () const
Get current logger mode. [More...](#)
- virtual DtTime **time** () const
Get/Set logger time. [More...](#)
- virtual void **setTime** (DtTime newTime)
- virtual DtTime **idealTime** () const
Get the ideal logger time. [More...](#)
- virtual void **skipTime** (DtTime amount)
Skip forward in simulation time. [More...](#)
- virtual DtLgrFilePosition **filePosition** () const
Get current file info: position, file length (in seconds), and logger file ptr. [More...](#)
- virtual DtTime **fileLength** () const
- virtual **LtLoggerFile*** **file** () const
- virtual void **gotoEnd** ()
Set whether we're at the beginning or end of the tape. [More...](#)
- virtual void **gotoStart** ()
- virtual double **timeScale** () const
Get/Set playback speed (default = 1.0). [More...](#)
- virtual void **setTimeScale** (double timescale)
- virtual DtBoolean **dataScaling** () const
Get/Set whether data scaling will take place. [More...](#)
- virtual void **setDataScaling** (DtBoolean onOff)
- virtual DtLgrMode **eofAction** () const
Get/Set the EOF action. [More...](#)
- virtual void **setEofAction** (DtLgrMode e)
- virtual DtLgrMode **playAction** () const
Get/Set the Play action. [More...](#)

- virtual void **setPlayAction** (DtLgrMode m)
- virtual DtLgrMode **recordAction** () const
Get/Set the Record Action. [More...](#)
- virtual void **setRecordAction** (DtLgrMode m)
- virtual const char* **currentFile** () const
Set file for playback or recording (Returns DtTrue if load is successful) Get the current filename. [More...](#)
- virtual DtBoolean **loadFileForPlayback** (char *filename)
- virtual DtBoolean **loadFileForRecording** (char *filename)
- virtual void **play** ()
*Control the logger's mode (will take effect the next time **tick**() is called). [More...](#)*
- virtual void **record** ()
- virtual void **stop** ()
- virtual void **pause** ()
- virtual void **setMirroring** (DtBoolean onOff)
Set whether "mirroring" is on A value of DtTrue causes all logger actions to produce logger control pdus/interactions. [More...](#)
- virtual void **setPacketLimit** (int val)
*Set a limit on the number of packets which can be generated by the logger pipeline during packet processing. The default is 1000 packets (and is defined in **lgrState.h**). A value of -1 indicates that there's no limit on the number of packets that get generated. This value does not effect the number of packets that actually get sent out on the network; Instead, it effects the number of packets that can processed internally at one time. [More...](#)*
- virtual void **setRecordingLoggerControlInfo** (DtBoolean onOff)
Get/Set whether logger control information should get recorded. [More...](#)
- virtual DtBoolean **recordingLoggerControlInfo** () const
- virtual void **setDefaultRecordSourceParameter** (DtString param)
Set the parameter to get sent to the default Record Source Object; The default parameter is "None". [More...](#)
- virtual DtBoolean **inPlayback** () const
Get whether we're in a playback state or recording state. [More...](#)
- virtual void **registerObject** (const char *keyname, **DtCreateMethod** m)
Register an object to the pipeline. [More...](#)
- virtual DtBoolean **addObject** (**LtObjectType** t, const char *keyname, DtString theParam)
Enable a registered object and add it to the pipeline. [More...](#)
- virtual DtBoolean **removeObject** (**LtObjectType** t, const char *keyname)
Disable a registered object and delete it from the pipeline. [More...](#)

- virtual void **setPlaybackFileCreator** (**DtLoggerPlaybackConstructMethod** m)
To over-ride the default playback and record file, register an alternate creation function with these functions. [More...](#)
- virtual void **setRecordFileCreator** (**DtLoggerRecordConstructMethod** m)
- virtual **DtAnnotationManager*** **annotations** () const
- virtual void **postToClipboard** (DtString keyname, void *data)
Make objects available to pipeline object through LgrState's "clipboard". [More...](#)
- virtual void* **fetchFromClipboard** (DtString keyname)
- virtual void* **removeFromClipboard** (DtString keyname)
- virtual void **reset** ()
Reset logger to start state; This function gets called before the logger opens new files for playback or recording. [More...](#)
- virtual double **heartBeat** () const
Get/Set the current heartbeat interval. [More...](#)
- virtual void **setHeartBeat** (double val)
- virtual double **timeout** () const
Get/Set the current timeout interval. [More...](#)
- virtual void **setTimeout** (double val)

Protected Methods

- void **init** ()
- void **setup** (char *)
init logger object. [More...](#)
- void **setMode** (DtLgrMode m)
setup logger for recording or playback. [More...](#)

Protected Attributes

- DtString **myFilename**
- DtString **myRecSrcParam**
- DtBoolean **myPb**
- DtBoolean **myMirror**
- DtBoolean **myCreatedExConn**

- DtBoolean **myRecLgrCtrlInfo**
 - double **myTimeScale**
 - DtExerciseConn* **myExConn**
 - **DtLgrPbClock*** **myClock**
 - DtLgrPipeline* **myPipeline**
 - **DtLgrState*** **myState**
 - double **myHeartbeat**
 - double **myTimeout**
 - DtString **myPbSource**
 - DtString **myPbSend**
 - DtString **myRecSource**
 - DtString **myRecSend**
-

Constructor & Destructor Documentation

```
DtLogger::DtLogger (DtExerciseConn * exConn )
```

Constructor.

```
DtLogger::DtLogger (int port,  
                    int exercise_id,  
                    int site_id,  
                    int application_num )
```

Constructors.

```
DtLogger::DtLogger (int port,  
                    DtInetAddr destAddr,  
                    int exercise_id,  
                    int site_id,  
                    int application_num )
```

```
DtLogger::DtLogger (DtSocket * skt,  
                    int exercise_id,  
                    int site_id,  
                    int application_num )
```

```
DtLogger::~DtLogger () [virtual]
```

Destructor.

Member Function Documentation

```
DtBoolean DtLogger::addObject (LtObjectType t,  
                               const char * keyname,  
                               DtString theParam ) [virtual]
```

Enable a registered object and add it to the pipeline.

```
DtAnnotationManager * DtLogger::annotations () const [virtual]
```

```
const char * DtLogger::currentFile () const [virtual]
```

Set file for playback or recording (Returns DtTrue if load is successful) Get the current filename.

```
DtBoolean DtLogger::dataScaling () const [virtual]
```

Get/Set whether data scaling will take place.

```
DtLgrMode DtLogger::eofAction () const [virtual]
```

Get/Set the EOF action.

DtExerciseConn * DtLogger::exerciseConn () const [virtual]

Get exercise conn.

void * DtLogger::fetchFromClipboard (DtString *keyname*) [virtual]

[LtLoggerFile](#) * DtLogger::file () const [virtual]

DtTime DtLogger::fileLength () const [virtual]

DtLgrFilePosition DtLogger::filePosition () const [virtual]

Get current file info: position, file length (in seconds), and logger file ptr.

void DtLogger::gotoEnd () [virtual]

Set whether we're at the beginning or end of the tape.

void DtLogger::gotoStart () [virtual]

double DtLogger::heartBeat () const [virtual]

Get/Set the current heartbeat interval.

DtTime DtLogger::idealTime () const [virtual]

Get the ideal logger time.

```
DtBoolean DtLogger::inPlayback () const [virtual]
```

Get whether we're in a playback state or recording state.

```
void DtLogger::init () [protected]
```

```
DtBoolean DtLogger::loadFileForPlayback (char * filename ) [virtual]
```

```
DtBoolean DtLogger::loadFileForRecording (char * filename ) [virtual]
```

```
DtLgrMode DtLogger::mode () const [virtual]
```

Get current logger mode.

```
void DtLogger::pause () [virtual]
```

```
void DtLogger::play () [virtual]
```

Control the logger's mode (will take effect the next time **tick()** is called).

```
DtLgrMode DtLogger::playAction () const [virtual]
```

Get/Set the Play action.

```
void DtLogger::postToClipboard (DtString keyname,  
                                void * data ) [virtual]
```

Make objects available to pipeline object through LgrState's "clipboard".

```
void DtLogger::record () [virtual]
```

```
DtLgrMode DtLogger::recordAction () const [virtual]
```

Get/Set the Record Action.

```
DtBoolean DtLogger::recordingLoggerControlInfo () const [virtual]
```

```
void DtLogger::registerObject (const char * keyname,  
                               DtCreateMethod m) [virtual]
```

Register an object to the pipeline.

```
void * DtLogger::removeFromClipboard (DtString keyname) [virtual]
```

```
DtBoolean DtLogger::removeObject (LtObjectType t,  
                                   const char * keyname) [virtual]
```

Disable a registered object and delete it from the pipeline.

```
void DtLogger::reset () [virtual]
```

Reset logger to start state; This function gets called before the logger opens new files for playback or recording.

```
void DtLogger::setDataScaling (DtBoolean onOff) [virtual]
```

```
void DtLogger::setDefaultRecordSourceParameter (DtString param) [virtual]
```

Set the parameter to get sent to the default Record Source Object; The default parameter is "None".

```
void DtLogger::setEofAction (DtLgrMode e ) [virtual]
```

```
void DtLogger::setHeartBeat (double val ) [virtual]
```

```
void DtLogger::setMirroring (DtBoolean onOff) [virtual]
```

Set whether "mirroring" is on A value of DtTrue causes all logger actions to produce logger control pdus/interactions.

```
void DtLogger::setMode (DtLgrMode m ) [protected]
```

setup logger for recording or playback.

```
void DtLogger::setPacketLimit (int val ) [virtual]
```

Set a limit on the number of packets which can be generated by the logger pipeline during packet processing. The default is 1000 packets (and is defined in [lgrState.h](#)). A value of -1 indicates that there's no limit on the number of packets that get generated. This value does not effect the number of packets that actually get sent out on the network; Instead, it effects the number of packets that can processed internally at one time.

```
void DtLogger::setPlayAction (DtLgrMode m ) [virtual]
```

```
void DtLogger::setPlaybackFileCreator (DtLoggerPlaybackConstructMethod m ) [virtual]
```

To over-ride the default playback and record file, register an alternate creation function with these functions.

```
void DtLogger::setRecordAction (DtLgrMode m) [virtual]
```

```
void DtLogger::setRecordFileCreator (DtLoggerRecordConstructMethod m) [virtual]
```

```
void DtLogger::setRecordingLoggerControlInfo (DtBoolean onOff) [virtual]
```

Get/Set whether logger control information should get recorded.

```
void DtLogger::setTime (DtTime newTime) [virtual]
```

```
void DtLogger::setTimeScale (double timescale) [virtual]
```

```
void DtLogger::setTimeout (double val) [virtual]
```

```
void DtLogger::setup (char *) [protected]
```

init logger object.

```
void DtLogger::skipTime (DtTime amount) [virtual]
```

Skip forward in simulation time.

```
void DtLogger::stop () [virtual]
```

DtTime DtLogger::tick () [virtual]

Updates pipeline elements, generates packets, and process packets.
Returns the arrival time of the next packet. If the arrival time is unknown, returns -1.

DtTime DtLogger::time () const [virtual]

Get/Set logger time.

double DtLogger::timeScale () const [virtual]

Get/Set playback speed (default = 1.0).

double DtLogger::timeout () const [virtual]

Get/Set the current timeout interval.

Member Data Documentation

[DtLgrPbClock](#) * DtLogger::myClock [protected]

DtBoolean DtLogger::myCreatedExConn [protected]

DtExerciseConn * DtLogger::myExConn [protected]

DtString DtLogger::myFilename [protected]

double DtLogger::myHeartbeat [protected]

DtBoolean DtLogger::myMirror [protected]

DtBoolean DtLogger::myPb [protected]

DtString DtLogger::myPbSend [protected]

DtString DtLogger::myPbSource [protected]

DtLgrPipeline * DtLogger::myPipeline [protected]

DtBoolean DtLogger::myRecLgrCtrlInfo [protected]

DtString DtLogger::myRecSend [protected]

DtString DtLogger::myRecSource [protected]

DtString DtLogger::myRecSrcParam [protected]

DtLgrState * DtLogger::myState [protected]

double DtLogger::myTimeScale [protected]

double DtLogger::myTimeout [protected]

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

- [logger.h](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

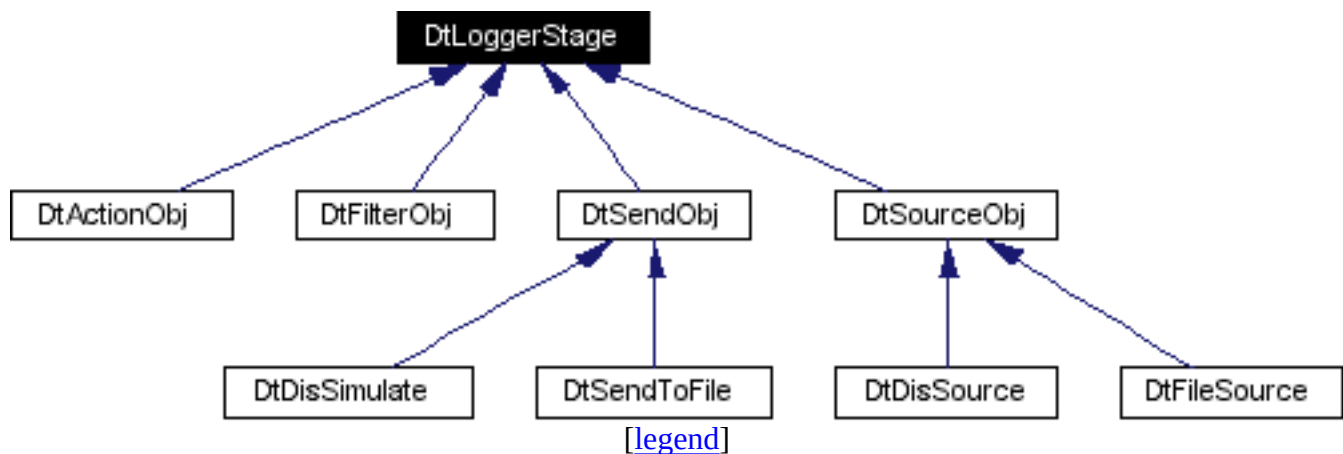
DtLoggerStage Class Reference

this is an abstract class that defines a logger pipeline Object for the logger agent to perform specific actions. examples can be ramToools interface. [More...](#)

```
#include <loggerStage.h>
```

Inherited by **DtActionObj**, **DtFilterObj**, **DtSendObj**, and **DtSourceObj**.

Inheritance diagram for DtLoggerStage:



[List of all members.](#)

Public Methods

- **DtLoggerStage** (const char *theName)
- virtual **~DtLoggerStage** ()
- virtual void **initSelf** (**DtLgrState** *theState)=0
this method is called once after construction to perform some initial initialization. [More...](#)
- virtual void **activate** ()=0
this method is called when the logger agent has selected to send data to this object. [More...](#)
- virtual void **deactivate** ()=0
this method is called when the logger agent is no longer going to send data to this object. [More...](#)

- virtual void **reset** ()=0
Reset object. [More...](#)
- virtual void **tick** ()=0
this method gives the object execution time. [More...](#)
- virtual DtBoolean **isOk** () const=0
- virtual const char* **getName** () const
- virtual const char* **printInfo** () const=0
this method returns a string that supplies identifying info about the object. [More...](#)

Protected Methods

- **DtLoggerStage** (const DtLoggerStage &)

Protected Attributes

- DtString **myName**
-

Detailed Description

this is an abstract class that defines a logger pipeline Object for the logger agent to perform specific actions. examples can be ramToools interface.

Constructor & Destructor Documentation

```
DtLoggerStage::DtLoggerStage (const char * theName )
```

```
DtLoggerStage::~DtLoggerStage () [virtual]
```

DtLoggerStage::DtLoggerStage (const DtLoggerStage &) [protected]

Member Function Documentation

void DtLoggerStage::activate () [pure virtual]

this method is called when the logger agent has selected to send data to this object.

Reimplemented in **DtDisSimulate**, **DtDisSource**, **DtFileSource**, **DtFilterObj**, **DtSendObj**, **DtSendToFile**, **DtSourceObj**, and **DtActionObj**.

void DtLoggerStage::deactivate () [pure virtual]

this method is called when the logger agent is no longer going to send data to this object.

Reimplemented in **DtDisSimulate**, **DtDisSource**, **DtFileSource**, **DtFilterObj**, **DtSendObj**, **DtSendToFile**, **DtSourceObj**, and **DtActionObj**.

const char * DtLoggerStage::getName () const [virtual]

Reimplemented in **DtFilterObj**, and **DtActionObj**.

void DtLoggerStage::initSelf (DtLgrState * *theState*) [pure virtual]

this method is called once after construction to perform some initial initialization.

Reimplemented in **DtDisSimulate**, **DtDisSource**, **DtFileSource**, **DtFilterObj**, **DtSendObj**, **DtSendToFile**, **DtSourceObj**, and **DtActionObj**.

DtBoolean DtLoggerStage::isOk () const [pure virtual]

Reimplemented in [DtFilterObj](#), [DtSendObj](#), [DtSourceObj](#), and [DtActionObj](#).

```
const char * DtLoggerStage::printInfo () const [pure virtual]
```

this method returns a string that supplies identifying info about the object.

Reimplemented in [DtDisSimulate](#), [DtDisSource](#), [DtFileSource](#), [DtFilterObj](#), [DtSendObj](#), [DtSendToFile](#), [DtSourceObj](#), and [DtActionObj](#).

```
void DtLoggerStage::reset () [pure virtual]
```

Reset object.

Reimplemented in [DtDisSource](#), [DtFileSource](#), [DtSendObj](#), [DtSourceObj](#), and [DtActionObj](#).

```
void DtLoggerStage::tick () [pure virtual]
```

this method gives the object execution time.

Reimplemented in [DtDisSimulate](#), [DtFileSource](#), [DtFilterObj](#), [DtSendObj](#), [DtSourceObj](#), and [DtActionObj](#).

Member Data Documentation

```
DtString DtLoggerStage::myName [protected]
```

Reimplemented in [DtSourceObj](#).

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

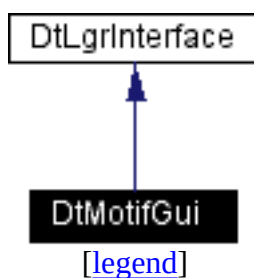
- [loggerStage.h](#)

DtMotifGui Class Reference

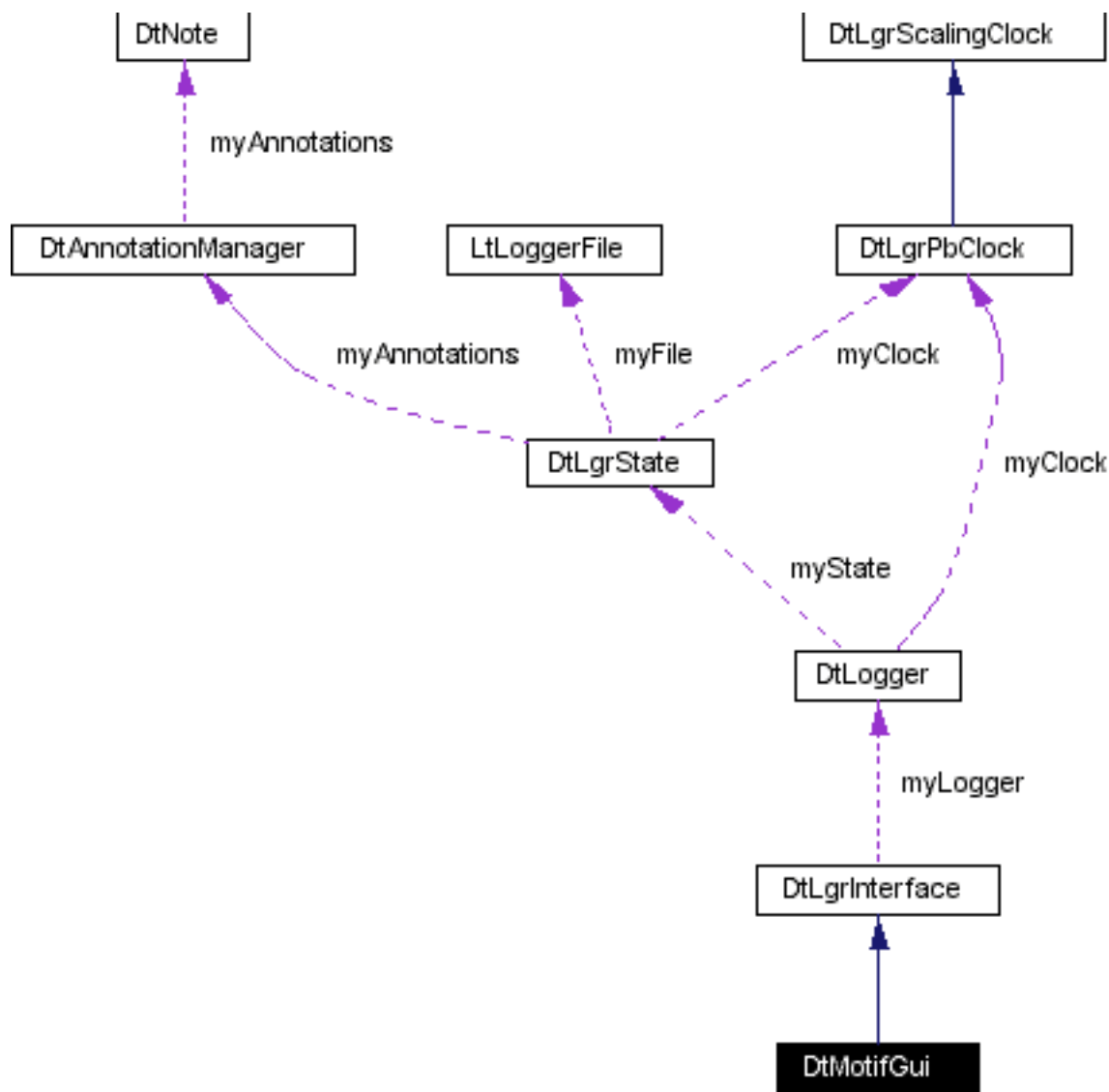
`#include <motifGui.h>`

Inherits **[DtLgrInterface](#)**.

Inheritance diagram for DtMotifGui:



Collaboration diagram for DtMotifGui:



[[legend](#)]

[List of all members.](#)

Public Methods

- **DtMotifGui** (**DtLogger** *logger, DtBoolean advanced=DtFalse)
- virtual ~**DtMotifGui** ()
- virtual void **tick** ()
- virtual void **resetAnnotList** ()
- virtual void **updateAnnotList** ()
- void **createInterface** (Widget top)
- int **getTsIndex** (float val)
- int **askYN** (char *prompt, int defChoice)
- void **initSelf** (**DtLogger** *theAIF)
- void **arrangeForMode** (DtLgrMode m)
- void **showTime** ()

- void **showMode** (DtLgrMode m)
- void **showFilename** (char *f)
- void **displayButtons** (DtLgrMode m)
- void **shadeAndSensitivizeButton** (Widget, Pixmap, Pixmap, Boolean)
- void **showPositional** ()
- void **showScale** ()
- void **colorWidget** (Widget w, int inverse)
- void **processAnnotationFile** (const char *theFile)
- void **warn** (char *warning)
- void **saveNoteList** (const char *filename)

Static Public Methods

- void **openCallback** (Widget, DtMotifGui *aif, XtPointer)
- void **newCallback** (Widget, DtMotifGui *aif, XtPointer)
- void **quitCallback** (Widget, DtMotifGui *aif, XtPointer)
- void **configCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **configRadioCallback** (Widget w, int which, XmToggleButtonCallbackStruct *state)
- void **setTimeCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **skipTimeCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **annotCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **timeScaleCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **helpCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **aboutCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **aboutOkCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **playCallback** (Widget, XtPointer, XtPointer)
- void **stopCallback** (Widget, XtPointer, XtPointer)
- void **pauseCallback** (Widget, XtPointer, XtPointer)
- void **recordCallback** (Widget, XtPointer, XtPointer)
- void **SOFCallback** (Widget, XtPointer, XtPointer)
- void **EOFCallback** (Widget, XtPointer, XtPointer)
- void **openFileOkCallback** (Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs)
- void **openFileCancelCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **newFileOkCallback** (Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs)
- void **newFileCancelCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **setTimeOkCallback** (Widget w, DtMotifGui *aif, XmSelectionBoxCallbackStruct *cbs)
- void **setTimeCancelCallback** (Widget w, DtMotifGui *aif, XtPointer)

- void **setTimeHelpCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **skipTimeOkCallback** (Widget w, DtMotifGui *aif, XmSelectionBoxCallbackStruct *cbs)
- void **skipTimeCancelCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **skipTimeHelpCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **setScaleOkCallback** (Widget w, DtMotifGui *aif, XmSelectionBoxCallbackStruct *c)
- void **setScaleCancelCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **setScaleHelpCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **configOkCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **configCancelCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **configHelpCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **skipTimeToggleCallback** (Widget w, int n, XmToggleButtonCallbackStruct *cbs)
- void **playActToggleCallback** (Widget w, int n, XmToggleButtonCallbackStruct *cbs)
- void **recordActToggleCallback** (Widget w, int n, XmToggleButtonCallbackStruct *cbs)
- void **EOFActToggleCallback** (Widget w, int n, XmToggleButtonCallbackStruct *cbs)
- void **annotOpenCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **annotImportCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **annotCloseCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **annotClearCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **openAnnotOkCallback** (Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs)
- void **openAnnotCancelCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **listSelectionCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **importAnnotOkCallback** (Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs)
- void **jumpCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **annotAddCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **annotModCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **annotRemoveCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **annotHelpMenuCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **saveasAnnotOkCallback** (Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs)
- void **annotSaveasCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **annotSaveCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **addAnnotOkCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **addAnnotHelpCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **modAnnotOkCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **modAnnotHelpCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **manInputCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **manOutputCallback** (Widget w, DtMotifGui *aif, XtPointer)

- void **addInputCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **addOutputCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **addInputOkCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **addOutputOkCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **addInputSelectCallback** (Widget w, void *num, XtPointer)
- void **addOutputSelectCallback** (Widget w, void *num, XtPointer)
- void **inputRemoveCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **outputRemoveCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **fileBrowseCallback** (Widget w, DtMotifGui *aif, XtPointer)
- void **fileBrowseOkCallback** (Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs)

Public Attributes

- Widget **topLevel**
- Widget **topManager**
- Widget **topForm**
- Widget **menuBar**
- Widget **fileMenu**
- Widget **openMenu**
- Widget **newMenu**
- Widget **optionMenu**
- Widget **setTimeMenu**
- Widget **skipTimeMenu**
- Widget **annotMenu**
- Widget **timeScaleMenu**
- Widget **helpMenu**
- Widget **currentTimeField**
- Widget **currentStatusField**
- Widget **currentFileField**
- Widget **stopButton**
- Widget **recordButton**
- Widget **playButton**
- Widget **pauseButton**
- Widget **SOFButton**
- Widget **EOFButton**
- Widget **stopLabel**
- Widget **recordLabel**

- Widget **playLabel**
- Widget **pauseLabel**
- Widget **SOFLabel**
- Widget **EOFLabel**
- Widget **currentTimeLabel**
- Widget **openFileWidget**
- Widget **newFileWidget**
- Widget **configWidget**
- Widget **playActPlayRadio**
- Widget **playActStopRadio**
- Widget **recordActRecordRadio**
- Widget **recordActStopRadio**
- Widget **eofActLoopRadio**
- Widget **eofActStopRadio**
- Widget **eofActPauseRadio**
- Widget **heartbeatWidget**
- Widget **timeoutWidget**
- Widget **setTimeWidget**
- Widget **setScaleWidget**
- Widget **skipTimeWidget**
- Widget **aboutWidget**
- Widget **annotWidget**
- Widget **annotList**
- Widget **annotText**
- Widget **annotCurFile**
- Widget **openAnnotWidget**
- Widget **importAnnotWidget**
- Widget **saveasAnnotWidget**
- Widget **addAnnotWidget**
- Widget **addAnnotTimeLabel**
- Widget **addAnnotTextWidget**
- Widget **modAnnotWidget**
- Widget **modAnnotTimeLabel**
- Widget **modAnnotTextWidget**
- Widget **manInputWidget**
- Widget **manInputMenu**
- Widget **inputList**
- Widget **manOutputWidget**

- Widget **manOutputMenu**
- Widget **outputList**
- Widget **addInputButton**
- Widget **addOutputButton**
- Widget **removeInputButton**
- Widget **removeOutputButton**
- Widget **addInputWidget**
- Widget **addOutputWidget**
- Widget **addParmTextWidget**
- Widget **addOutputTextWidget**
- Widget **addBrowseWidget**
- Widget **addOutputBrowseWidget**
- Widget **fileBrowseWidget**
- Widget **playToggleBox**
- Widget **openActToggleBox**
- Widget **shell**
- int **inputSelNum**
- int **outputSelNum**
- int **mySelectedAnnotation**
- TimeScaleSlider* **timeScaleSlider**
- Pixmap **playPix**
- Pixmap **playPixG**
- Pixmap **stopPix**
- Pixmap **stopPixG**
- Pixmap **pausePix**
- Pixmap **pausePixG**
- Pixmap **recordPix**
- Pixmap **recordPixG**
- Pixmap **recordPixPG**
- Pixmap **playPixP**
- Pixmap **pausePixPG**
- Pixmap **recordPixP**
- Pixmap **playPixPG**
- Pixmap **sofPix**
- Pixmap **eofPix**
- Pixmap **sofPixG**
- Pixmap **eofPixG**
- Pixmap **jumpPix**

- QPixmap **editPix**
- QPixmap **logoPix**
- XtAppContext **context**
- DtLgrMode **pendingEOFAction**
- DtLgrMode **pendingPlayAction**
- DtLgrMode **pendingRecordAction**
- DtLgrMode **cachedMode**
- double **myAnnotAddTime**
- int **myAdvanced**
- DtString **myBrowser**
- int **browserOpen**
- char **cachedTime** [80]
- char **cachedScale** [80]
- int **bmArray** [6][2]
- int **buttonActive** [6]
- int **buttonPressed** [6]
- float* **timeScaleValues**
- int **timeScaleValuesCount**
- int **pb**
- int **myAnnotCount**
- DtTime **annotGuiMapping** [MAX_ANNOTS]

Constructor & Destructor Documentation

```
DtMotifGui::DtMotifGui (DtLogger * logger,  
                        DtBoolean advanced = DtFalse )
```

```
DtMotifGui::~~DtMotifGui () [virtual]
```

Member Function Documentation


```
void DtMotifGui::configHelpCallback (Widget w,  
                                     DtMotifGui * aif,  
                                     XtPointer ) [static]
```

```
void DtMotifGui::configOkCallback (Widget w,  
                                   DtMotifGui * aif,  
                                   XtPointer ) [static]
```

```
void DtMotifGui::configRadioCallback (Widget w,  
                                       int which,  
                                       XmToggleButtonCallbackStruct * state ) [static]
```

```
void DtMotifGui::createInterface (Widget top )
```

```
void DtMotifGui::displayButtons (DtLgrMode m )
```

```
void DtMotifGui::fileBrowseCallback (Widget w,  
                                     DtMotifGui * aif,  
                                     XtPointer ) [static]
```

```
void DtMotifGui::fileBrowseOkCallback (Widget w,  
                                       DtMotifGui * aif,  
                                       XmFileSelectionBoxCallbackStruct * cbs ) [static]
```

```
int DtMotifGui::getTsIndex (float val )
```



```
void DtMotifGui::manOutputCallback (Widget w,  
                                   DtMotifGui * aif,  
                                   XtPointer ) [static]
```

```
void DtMotifGui::modAnnotHelpCallback (Widget w,  
                                       DtMotifGui * aif,  
                                       XtPointer ) [static]
```

```
void DtMotifGui::modAnnotOkCallback (Widget w,  
                                     DtMotifGui * aif,  
                                     XtPointer) [static]
```

```
void DtMotifGui::newCallback (Widget,
                             DtMotifGui * aif,
                             XtPointer ) [static]
```

[illegible]

```
void DtMotifGui::newFileOkCallback (Widget w,  
                                   DtMotifGui * aif,  
                                   XmFileSelectionBoxCallbackStruct * cbs ) [static]
```

[illegible]

```
void DtMotifGui::openAnnotOkCallback (Widget w,  
                                       DtMotifGui * aif,  
                                       XmFileSelectionBoxCallbackStruct * cbs) [static]
```

```
void DtMotifGui::openCallback (Widget,  
                               DtMotifGui * aif,  
                               XtPointer) [static]
```

```
void DtMotifGui::openFileCancelCallback (Widget w,  
                                          DtMotifGui * aif,  
                                          XtPointer) [static]
```

```
void DtMotifGui::openFileOkCallback (Widget w,  
                                     DtMotifGui * aif,  
                                     XmFileSelectionBoxCallbackStruct * cbs) [static]
```

```
void DtMotifGui::outputRemoveCallback (Widget w,  
                                       DtMotifGui * aif,  
                                       XtPointer) [static]
```

```
void DtMotifGui::pauseCallback (Widget,  
                                XtPointer,  
                                XtPointer) [static]
```

```
void DtMotifGui::playActToggleCallback (Widget w,  
                                         int n,  
                                         XmToggleButtonCallbackStruct * cbs) [static]
```

```
void DtMotifGui::playCallback (Widget,  
                                XtPointer,  
                                XtPointer ) [static]
```

```
void DtMotifGui::processAnnotationFile (const char * theFile )
```

```
void DtMotifGui::quitCallback (Widget,  
                                DtMotifGui * aif,  
                                XtPointer ) [static]
```

```
void DtMotifGui::recordActToggleCallback (Widget w,  
                                           int n,  
                                           XmToggleButtonCallbackStruct * cbs ) [static]
```

```
void DtMotifGui::recordCallback (Widget,  
                                 XtPointer,  
                                 XtPointer ) [static]
```

```
void DtMotifGui::resetAnnotList () [virtual]
```

```
void DtMotifGui::saveNoteList (const char * filename )
```

```
void DtMotifGui::saveasAnnotOkCallback (Widget w,  
                                         DtMotifGui * aif,  
                                         XmFileSelectionBoxCallbackStruct * cbs )  
                                         [static]
```

```
void DtMotifGui::setScaleCancelCallback (Widget w,  
                                         DtMotifGui * aif,  
                                         XtPointer ) [static]
```

```
void DtMotifGui::setScaleHelpCallback (Widget w,  
                                       DtMotifGui * aif,  
                                       XtPointer ) [static]
```

```
void DtMotifGui::setScaleOkCallback (Widget w,  
                                     DtMotifGui * aif,  
                                     XmSelectionBoxCallbackStruct * c ) [static]
```

```
void DtMotifGui::setTimeCallback (Widget w,  
                                  DtMotifGui * aif,  
                                  XtPointer ) [static]
```

```
void DtMotifGui::setTimeCancelCallback (Widget w,  
                                         DtMotifGui * aif,  
                                         XtPointer ) [static]
```

```
void DtMotifGui::setTimeHelpCallback (Widget w,  
                                      DtMotifGui * aif,  
                                      XtPointer ) [static]
```

```
void DtMotifGui::setTimeOkCallback (Widget w,  
                                    DtMotifGui * aif,  
                                    XmSelectionBoxCallbackStruct * cbs ) [static]
```



```
void DtMotifGui::skipTimeOkCallback (Widget w,  
                                     DtMotifGui * aif,  
                                     XmSelectionBoxCallbackStruct * cbs) [static]
```

```
void DtMotifGui::skipTimeToggleCallback (Widget w,  
                                          int n,  
                                          XmToggleButtonCallbackStruct * cbs) [static]
```

```
void DtMotifGui::stopCallback (Widget,  
                               XtPointer,  
                               XtPointer) [static]
```

```
void DtMotifGui::tick () [virtual]
```

Reimplemented from **DtLgrInterface**.

```
void DtMotifGui::timeScaleCallback (Widget w,  
                                    DtMotifGui * aif,  
                                    XtPointer) [static]
```

```
void DtMotifGui::updateAnnotList () [virtual]
```

```
void DtMotifGui::warn (char * warning)
```

Member Data Documentation

```
Widget DtMotifGui::EOFButton
```

Widget DtMotifGui::EOFLabel

Widget DtMotifGui::SOFButton

Widget DtMotifGui::SOFLabel

Widget DtMotifGui::aboutWidget

Widget DtMotifGui::addAnnotTextWidget

Widget DtMotifGui::addAnnotTimeLabel

Widget DtMotifGui::addAnnotWidget

Widget DtMotifGui::addBrowseWidget

Widget DtMotifGui::addInputButton

Widget DtMotifGui::addInputWidget

Widget DtMotifGui::addOutputBrowseWidget

Widget DtMotifGui::addOutputButton

Widget DtMotifGui::addOutputTextWidget

Widget DtMotifGui::addOutputWidget

Widget DtMotifGui::addParmTextWidget

Widget DtMotifGui::annotCurFile

DtTime DtMotifGui::annotGuiMapping

Widget DtMotifGui::annotList

Widget DtMotifGui::annotMenu

Widget DtMotifGui::annotText

Widget DtMotifGui::annotWidget

int DtMotifGui::bmArray

int DtMotifGui::browserOpen

int DtMotifGui::buttonActive

int DtMotifGui::buttonPressed

DtLgrMode DtMotifGui::cachedMode

char DtMotifGui::cachedScale

char DtMotifGui::cachedTime

Widget DtMotifGui::configWidget

XtAppContext DtMotifGui::context

Widget DtMotifGui::currentFileField

Widget DtMotifGui::currentStatusField

Widget DtMotifGui::currentTimeField

Widget DtMotifGui::currentTimeLabel

Pixmap DtMotifGui::editPix

Widget DtMotifGui::eofActLoopRadio

Widget DtMotifGui::eofActPauseRadio

Widget DtMotifGui::eofActStopRadio

Pixmap DtMotifGui::eofPix

Pixmap DtMotifGui::eofPixG

Widget DtMotifGui::fileBrowseWidget

Widget DtMotifGui::fileMenu

Widget DtMotifGui::heartbeatWidget

Widget DtMotifGui::helpMenu

Widget DtMotifGui::importAnnotWidget

Widget DtMotifGui::inputList

int DtMotifGui::inputSelNum

Pixmap DtMotifGui::jumpPix

Pixmap DtMotifGui::logoPix

Widget DtMotifGui::manInputMenu

Widget DtMotifGui::manInputWidget

Widget DtMotifGui::manOutputMenu

Widget DtMotifGui::manOutputWidget

Widget DtMotifGui::menuBar

Widget DtMotifGui::modAnnotTextWidget

Widget DtMotifGui::modAnnotTimeLabel

Widget DtMotifGui::modAnnotWidget

int DtMotifGui::myAdvanced

double DtMotifGui::myAnnotAddTime

int DtMotifGui::myAnnotCount

DtString DtMotifGui::myBrowser

int DtMotifGui::mySelectedAnnotation

Widget DtMotifGui::newFileWidget

Widget DtMotifGui::newMenu

Widget DtMotifGui::openActToggleBox

Widget DtMotifGui::openAnnotWidget

Widget DtMotifGui::openFileWidget

Widget DtMotifGui::openMenu

Widget DtMotifGui::optionMenu

Widget DtMotifGui::outputList

int DtMotifGui::outputSelNum

Widget DtMotifGui::pauseButton

Widget DtMotifGui::pauseLabel

Pixmap DtMotifGui::pausePix

Pixmap DtMotifGui::pausePixG

Pixmap DtMotifGui::pausePixPG

int DtMotifGui::pb

DtLgrMode DtMotifGui::pendingEOFAction

DtLgrMode DtMotifGui::pendingPlayAction

DtLgrMode DtMotifGui::pendingRecordAction

Widget DtMotifGui::playActPlayRadio

Widget DtMotifGui::playActStopRadio

Widget DtMotifGui::playButton

Widget DtMotifGui::playLabel

Pixmap DtMotifGui::playPix

Pixmap DtMotifGui::playPixG

Pixmap DtMotifGui::playPixP

Pixmap DtMotifGui::playPixPG

Widget DtMotifGui::playToggleBox

Widget DtMotifGui::recordActRecordRadio

Widget DtMotifGui::recordActStopRadio

Widget DtMotifGui::recordButton

Widget DtMotifGui::recordLabel

Pixmap DtMotifGui::recordPix

Pixmap DtMotifGui::recordPixG

Pixmap DtMotifGui::recordPixP

Pixmap DtMotifGui::recordPixPG

Widget DtMotifGui::removeInputButton

Widget DtMotifGui::removeOutputButton

Widget DtMotifGui::saveasAnnotWidget

Widget DtMotifGui::setScaleWidget

Widget DtMotifGui::setTimeMenu

Widget DtMotifGui::setTimeWidget

Widget DtMotifGui::shell

Widget DtMotifGui::skipTimeMenu

Widget DtMotifGui::skipTimeWidget

Pixmap DtMotifGui::sofPix

Pixmap DtMotifGui::sofPixG

Widget DtMotifGui::stopButton

Widget DtMotifGui::stopLabel

Pixmap DtMotifGui::stopPix

Pixmap DtMotifGui::stopPixG

Widget DtMotifGui::timeScaleMenu

TimeScaleSlider * DtMotifGui::timeScaleSlider

float * DtMotifGui::timeScaleValues

int DtMotifGui::timeScaleValuesCount

Widget DtMotifGui::timeoutWidget

Widget DtMotifGui::topForm

Widget DtMotifGui::topLevel

Widget DtMotifGui::topManager

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

- **[motifGui.h](#)**

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtNote Class Reference

```
#include <note.h>
```

[List of all members.](#)

Public Methods

- **DtNote** (DtTime theTime, DtString theMsg)
- virtual void **setTime** (DtTime t)
- virtual DtTime **time** ()
- virtual void **setMsg** (char *m)
- virtual DtString& **msg** ()

Constructor & Destructor Documentation

```
DtNote::DtNote (DtTime theTime,  
                DtString theMsg) [inline]
```

Member Function Documentation

```
DtString & DtNote::msg () [inline, virtual]
```

```
void DtNote::setMsg (char * m) [inline, virtual]
```

```
void DtNote::setTime (DtTime t) [inline, virtual]
```

```
DtTime DtNote::time () [inline, virtual]
```

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

- [note.h](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtSendObj Class Reference

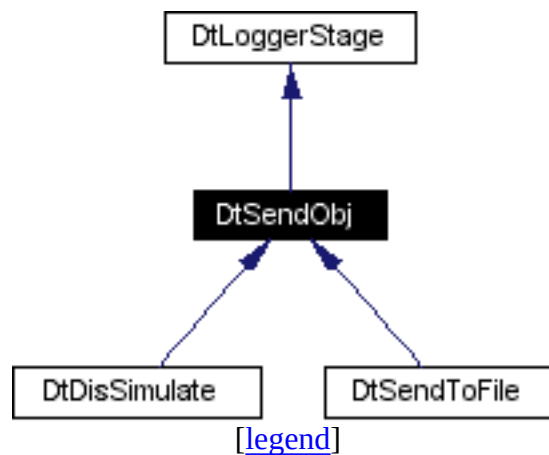
this is an abstract class that defines a target for the logger agent to send data. examples can be files, databases or a hla federation. [More...](#)

```
#include <sendObj.h>
```

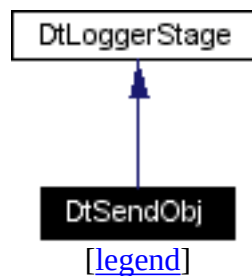
Inherits **DtLoggerStage**.

Inherited by **DtDisSimulate**, and **DtSendToFile**.

Inheritance diagram for DtSendObj:



Collaboration diagram for DtSendObj:



[List of all members.](#)

Public Methods

- **DtSendObj** (const char *theName)
- virtual **~DtSendObj** ()
- virtual void **initSelf** (**DtLgrState** *theState)=0
this method is called once after construction to perform some initial initialization. [More...](#)
- virtual void **activate** ()=0
this method is called when the logger agent has selected to send data to this object. [More...](#)
- virtual void **deactivate** ()=0
this method is called when the logger agent is no longer going to send data to this object. [More...](#)
- virtual void **reset** ()
Reset object. [More...](#)
- virtual void **processData** (const DtList *packetList)
this method is how the logger will send data to the object by default this method will iterate through the list and call sendPacket on each packet. [More...](#)
- virtual void **tick** ()
this method gives the object execution time. [More...](#)
- virtual DtBoolean **isOk** () const
- virtual const char* **printInfo** () const=0
this method returns a string that supplies identifying info about the object. [More...](#)
- virtual void **sendPacket** (const **LtPacketInfo** *packet)=0
this method is where the send object performs most of its work. [More...](#)

Protected Attributes

- DtBoolean **myStatus**

Detailed Description

this is an abstract class that defines a target for the logger agent to send data. examples can be files, databases or a hla federation.

Constructor & Destructor Documentation

`DtSendObj::DtSendObj (const char * theName )`

`DtSendObj::~DtSendObj () [virtual]`

Member Function Documentation

`void DtSendObj::activate () [pure virtual]`

this method is called when the logger agent has selected to send data to this object.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSimulate**, and **DtSendToFile**.

`void DtSendObj::deactivate () [pure virtual]`

this method is called when the logger agent is no longer going to send data to this object.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSimulate**, and **DtSendToFile**.

`void DtSendObj::initSelf (DtLgrState * theState ) [pure virtual]`

this method is called once after construction to perform some initial initialization.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSimulate**, and **DtSendToFile**.

DtBoolean DtSendObj::isOk () const [virtual]

Reimplemented from **DtLoggerStage**.

const char * DtSendObj::printInfo () const [pure virtual]

this method returns a string that supplies identifying info about the object.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSimulate**, and **DtSendToFile**.

void DtSendObj::processData (const DtList * *packetList*) [virtual]

this method is how the logger will send data to the object by default this method will iterate through the list and call sendPacket on each packet.

Reimplemented in **DtDisSimulate**, and **DtSendToFile**.

void DtSendObj::reset () [inline, virtual]

Reset object.

Reimplemented from **DtLoggerStage**.

void DtSendObj::sendPacket (const LtPacketInfo * *packet*) [pure virtual]

this method is where the send object performs most of its work.

Reimplemented in **DtDisSimulate**, and **DtSendToFile**.

void DtSendObj::tick () [inline, virtual]

this method gives the object execution time.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSimulate**.

Member Data Documentation

DtBoolean DtSendObj::myStatus [protected]

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

- [sendObj.h](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

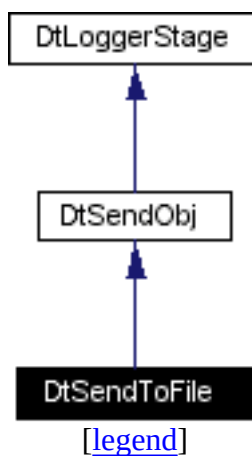
DtSendToFile Class Reference

For DT_HAVE_UNISTD_H this class will send a packet to a file. [More...](#)

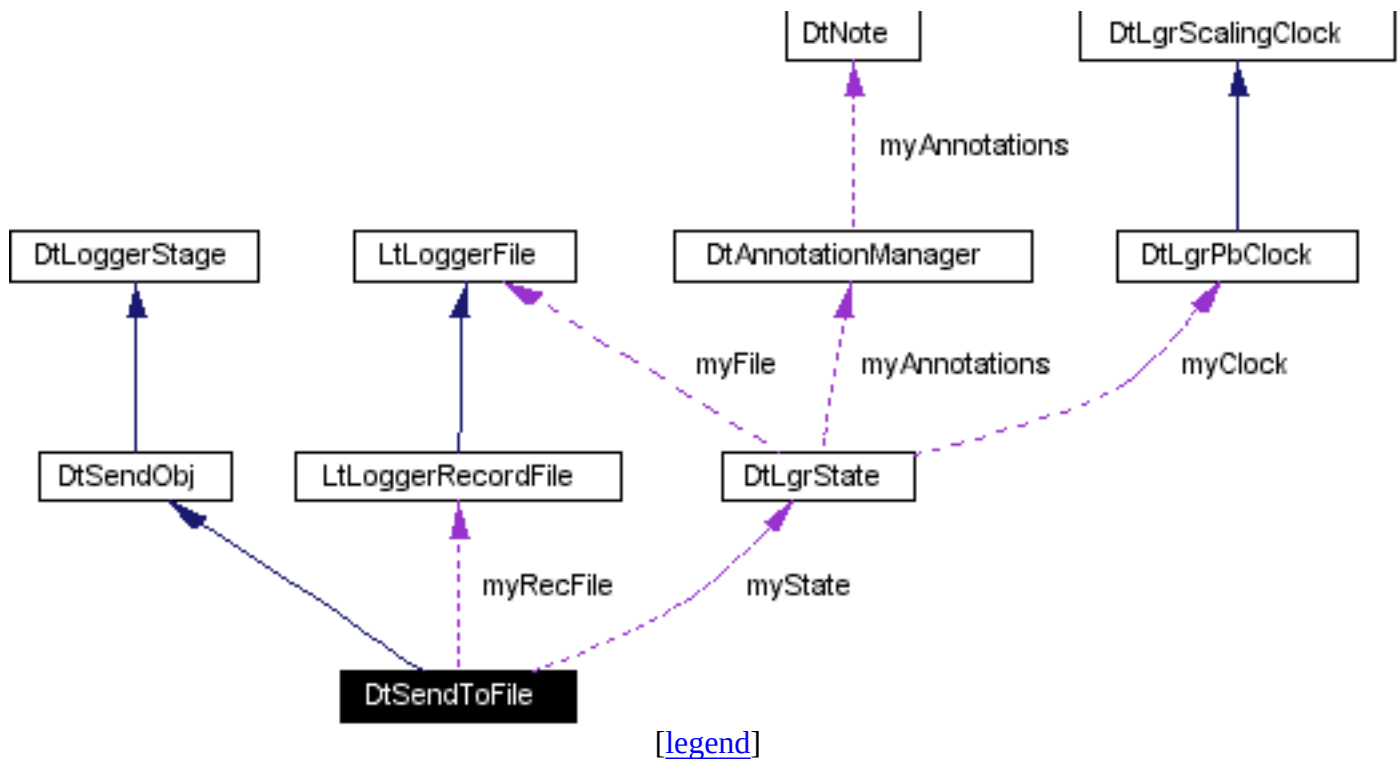
```
#include <sendToFile.h>
```

Inherits **DtSendObj**.

Inheritance diagram for DtSendToFile:



Collaboration diagram for DtSendToFile:



[List of all members.](#)

Public Methods

- **DtSendToFile** (DtString theName)
- **~DtSendToFile** ()
- virtual void **initSelf** (**DtLgrState** *theState)
this method is called once after construction to perform some initial initialization. [More...](#)
- virtual void **activate** ()
this method is called when the logger agent has selected to send data to this object. [More...](#)
- virtual void **deactivate** ()
this method is called when the logger agent is no longer going to send data to this object. [More...](#)
- virtual void **processData** (const DtList *packetList)
this method is how the logger will send data to the object. [More...](#)
- virtual void **sendPacket** (const **LtPacketInfo** *packet)
this method is where the send object performs most of its work. [More...](#)
- virtual const char* **printInfo** () const
this method returns a string that supplies identifying info about the object. [More...](#)

Static Public Methods

- void* **create** (DtString)

Protected Attributes

- FILE* **myFp**
this method writes the fom to the file. [More...](#)
 - DtString **myFileName**
 - **DtLgrState*** **myState**
 - **LtLoggerRecordFile*** **myRecFile**
-

Detailed Description

For DT_HAVE_UNISTD_H this class will send a packet to a file.

Constructor & Destructor Documentation

DtSendToFile::DtSendToFile (DtString *theName*)

DtSendToFile::~~DtSendToFile ()

Member Function Documentation

void DtSendToFile::activate () [virtual]

this method is called when the logger agent has selected to send data to this object.

Reimplemented from **DtSendObj**.

```
void * DtSendToFile::create (DtString) [static]
```

```
void DtSendToFile::deactivate () [virtual]
```

this method is called when the logger agent is no longer going to send data to this object.

Reimplemented from **DtSendObj**.

```
void DtSendToFile::initSelf (DtLgrState * theState) [virtual]
```

this method is called once after construction to perform some initial initialization.

Reimplemented from **DtSendObj**.

```
const char * DtSendToFile::printInfo () const [virtual]
```

this method returns a string that supplies identifying info about the object.

Reimplemented from **DtSendObj**.

```
void DtSendToFile::processData (const DtList * packetList) [virtual]
```

this method is how the logger will send data to the object.

Reimplemented from **DtSendObj**.

```
void DtSendToFile::sendPacket (const LtPacketInfo * packet) [inline, virtual]
```

this method is where the send object performs most of its work.

Reimplemented from **DtSendObj**.

Member Data Documentation

DtString DtSendToFile::myFileName [protected]

FILE * DtSendToFile::myFp [protected]

this method writes the fom to the file.

LtLoggerRecordFile * DtSendToFile::myRecFile [protected]

DtLgrState * DtSendToFile::myState [protected]

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

- [**sendToFile.h**](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtSourceObj Class Reference

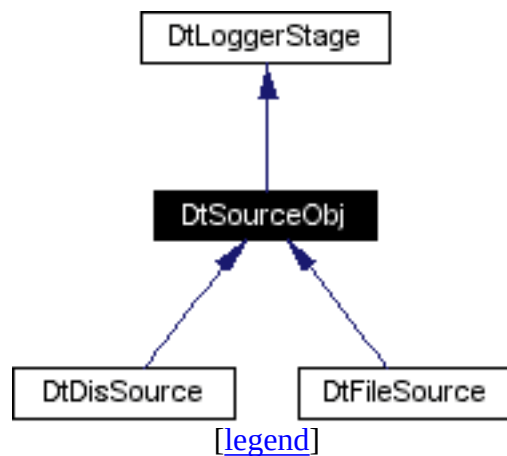
this is an abstract class that defines a data source for a logger agent. A source could be a HLA connection, TCP connection, etc. Each connection will override these virtual functions and produce conforming data to enter the data stream. [More...](#)

```
#include <sourceObj.h>
```

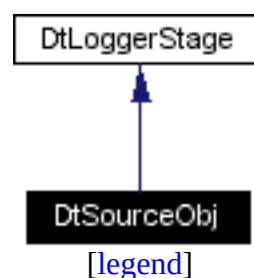
Inherits **DtLoggerStage**.

Inherited by **DtDisSource**, and **DtFileSource**.

Inheritance diagram for DtSourceObj:



Collaboration diagram for DtSourceObj:



[List of all members.](#)

Public Methods

- **DtSourceObj** (const char *theName)
- virtual **~DtSourceObj** ()
- virtual void **initSelf** (**DtLgrState** *theState)=0
this method is called soon after object creation to do begining initialization. [More...](#)
- virtual DtTime **activate** (DtTime startTime)=0
the Logger will call this method when it wishes to receive data from this source, returns when the first packet will be available relative to start time. negative if unknown. packets in the past should be skipped. [More...](#)
- virtual void **activate** ()
this method is called when the logger agent has selected to send data to this object. [More...](#)
- virtual void **deactivate** ()=0
The Logger will call this method when it no longer wishes to receive data from this source. [More...](#)
- virtual void **reset** ()=0
this method is called to indicate the object should reset itself to its initial state. Eg a file should return to the beginning. [More...](#)
- virtual DtTime **generatePackets** (DtTime currentTime, DtList *packetList)
The Logger calls this method to retrieve data from the source. the packet list is the current buffer of unprocessed packets from all sources. This method by default will repeatedly call generate packet until it returns NULL and add the packets to the list. The object should process all packets it has ready. The return value is an indication of when the next packet will be ready. sources such as data files should return the time for the next packet in the file. Real time sources should return negative to indicate the time of the next packet is unknown. [More...](#)
- virtual **LtPacketInfo*** **generatePacket** (DtTime currentTime, DtTime *nextTime)=0
this method returns one new packet that is ready. if no packet is ready it returns NULL. The time of the next packet is returned through nextTime parameter. [More...](#)
- virtual void **tick** ()
this method is called during the tick period in order for the source to have a chance to perform extra processing. [More...](#)
- virtual const char* **printInfo** () const
this method returns a string that contains identifying info about the object. [More...](#)
- virtual DtBoolean **isOk** () const

Protected Attributes

- DtString **myName**
- DtBoolean **myStatus**

Detailed Description

this is an abstract class that defines a data source for a logger agent. A source could be a HLA connection, TCP connection, etc. Each connection will override these virtual functions and produce conforming data to enter the data stream.

Constructor & Destructor Documentation

`DtSourceObj::DtSourceObj (const char * theName )`

`DtSourceObj::~~DtSourceObj () [virtual]`

Member Function Documentation

`void DtSourceObj::activate () [inline, virtual]`

this method is called when the logger agent has selected to send data to this object.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSource**, and **DtFileSource**.

`DtTime DtSourceObj::activate (DtTime startTime ) [pure virtual]`

the Logger will call this method when it wishes to receive data from this source, returns when the first packet will be available relative to start time. negative if unknown. packets in the past should be skipped.

Reimplemented in **DtDisSource**, and **DtFileSource**.

```
void DtSourceObj::deactivate() [pure virtual]
```

The Logger will call this method when it no longer wishes to receive data from this source.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSource**, and **DtFileSource**.

```
LtPacketInfo * DtSourceObj::generatePacket (DtTime currentTime,  
                                             DtTime * nextTime) [pure virtual]
```

this method returns one new packet that is ready. if no packet is ready it returns NULL. The time of the next packet is returned through nextTime parameter.

Reimplemented in **DtDisSource**, and **DtFileSource**.

```
DtTime DtSourceObj::generatePackets (DtTime currentTime,  
                                     DtList * packetList) [virtual]
```

The Logger calls this method to retrieve data from the source. the packet list is the current buffer of unprocessed packets from all sources. This method by default will repeatedly call generate packet until it returns NULL and add the packets to the list. The object should process all packets it has ready. The return value is an indication of when the next packet will be ready. sources such as data files should return the time for the next packet in the file. Real time sources should return negative to indicate the time of the next packet is unknown.

Reimplemented in **DtDisSource**, and **DtFileSource**.

```
void DtSourceObj::initSelf (DtLgrState * theState) [pure virtual]
```

this method is called soon after object creation to do begining initialization.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSource**, and **DtFileSource**.

DtBoolean DtSourceObj::isOk () const [virtual]

Reimplemented from **DtLoggerStage**.

const char * DtSourceObj::printInfo () const [virtual]

this method returns a string that contains identifying info about the object.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSource**, and **DtFileSource**.

void DtSourceObj::reset () [pure virtual]

this method is called to indicate the object should reset itself to its initial state. Eg a file should return to the beginning.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtDisSource**, and **DtFileSource**.

void DtSourceObj::tick () [inline, virtual]

this method is called during the tick period in order for the source to have a chance to perform extra processing.

Reimplemented from **DtLoggerStage**.

Reimplemented in **DtFileSource**.

Member Data Documentation

DtString DtSourceObj::myName [protected]

Reimplemented from **DtLoggerStage**.

DtBoolean DtSourceObj::myStatus [protected]

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

- **[sourceObj.h](#)**

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

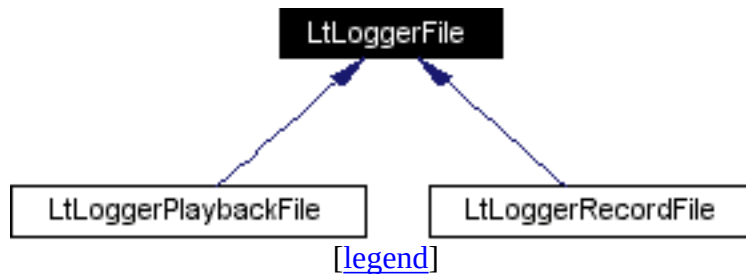
LtLoggerFile Class Reference

For DT_HAVE_UNISTD_H. [More...](#)

```
#include <loggerFile.h>
```

Inherited by **LtLoggerPlaybackFile**, and **LtLoggerRecordFile**.

Inheritance diagram for LtLoggerFile:



[List of all members.](#)

Public Methods

- virtual **~LtLoggerFile** ()
Destructor. [More...](#)
- virtual int **isOk** () const
Returns 1 if file has been opened successfully, 0 otherwise. [More...](#)
- virtual const char* **name** () const
Returns file name of the file opened. [More...](#)
- virtual FILE* **filePtr** () const

Protected Methods

- **LtLoggerFile** (const char *fileName, const char *flags)
Constructor. [More...](#)

- virtual FILE* **openFile** (const char *fileName, const char *flags)
Opens the file. [More...](#)
- virtual int **closeFile** ()
Closes the file. [More...](#)

Protected Attributes

- FILE* **myFp**
- int **myStatus**
- char **myFileName** [80]

Detailed Description

For DT_HAVE_UNISTD_H.

Constructor & Destructor Documentation

LtLoggerFile::~~LtLoggerFile () [virtual]

Destructor.

LtLoggerFile::LtLoggerFile (const char * *fileName*,
const char * *flags*) [protected]

Constructor.

Member Function Documentation

int LtLoggerFile::closeFile () [protected, virtual]

Closes the file.

```
FILE * LtLoggerFile::filePtr () const [virtual]
```

```
int LtLoggerFile::isOk () const [virtual]
```

Returns 1 if file has been opened successfully, 0 otherwise.

```
const char * LtLoggerFile::name () const [virtual]
```

Returns file name of the file opened.

```
FILE * LtLoggerFile::openFile (const char * fileName,  
                               const char * flags) [protected, virtual]
```

Opens the file.

Member Data Documentation

```
char LtLoggerFile::myFileName [protected]
```

```
FILE * LtLoggerFile::myFp [protected]
```

```
int LtLoggerFile::myStatus [protected]
```

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

- [loggerFile.h](#)

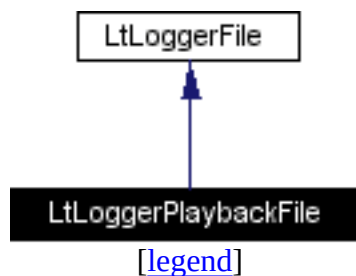
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

LtLoggerPlaybackFile Class Reference

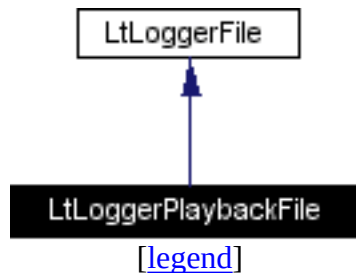
```
#include <loggerPbFile.h>
```

Inherits **LtLoggerFile**.

Inheritance diagram for LtLoggerPlaybackFile:



Collaboration diagram for LtLoggerPlaybackFile:



[List of all members.](#)

Public Methods

- **LtLoggerPlaybackFile** (const char *FileName)
Constructor. [More...](#)
- virtual ~**LtLoggerPlaybackFile** ()
Destructor. [More...](#)
- virtual **LtPacketInfo*** **readPacket** ()
Reads and returns next packet, NULL if none. In HLA, packets might be skipped if the file was created with a non-NULL exConn, and the packet cannot be converted to the exConn's FOM. [More...](#)

- virtual **LtPacketInfo*** **readPacketBackwards** ()
Reads and returns previous packet, NULL if none. In HLA, packets might be skipped if the file was created with a non-NULL exConn, and the packet cannot be converted to the exConn's FOM. [More...](#)
- virtual int **skipPacketForwards** ()
Skip over next packet either forwards or backwards direction. Return 1 if successful, 0 if no packets in that direction. [More...](#)
- virtual int **skipPacketBackwards** ()
- virtual int **jumpToStart** ()
Go to beginning/end of file. [More...](#)
- virtual int **jumpToEnd** ()
- virtual int **atStart** ()
Returns 1 if at beginning/end of file, 0 otherwise. [More...](#)
- virtual int **atEnd** ()
- virtual int **fileTimeBackwards** (DtTime *t)
Fills t with time of next or previous packet. Returns 1 on success, 0 if no packet in that direction. [More...](#)
- virtual int **fileTimeForwards** (DtTime *t)
- virtual int **firstPacketTime** (DtTime *t)
Fills t with time of first/last packet in file. [More...](#)
- virtual int **lastPacketTime** (DtTime *t)
- virtual void **seekTime** (DtTime t)
Jump to time t (in seconds) in file. [More...](#)
- virtual void **seekForward** (DtTime t)
Jump forward/backwards in file by t seconds. [More...](#)
- virtual void **seekBackward** (DtTime t)

Static Public Methods

- int **checkMagicNumber** (const char *filename)
This function checks filename's logger format Returns 1 if the file is valid; Prints error message and returns 0 otherwise. [More...](#)

Protected Methods

- virtual int **checkMagicNumber** ()
Reads the magic number at beginning of file. Returns 1 if it matches the expected version, 0 otherwise. [More...](#)
- virtual int **readPacketHeader** (**LtNetPacketHeader** *header)
Read header, data and tail into arguments. Return 1 if successful, 0 otherwise. [More...](#)
- virtual int **readPacketData** (void *buffer, int bytesToRead)
- virtual int **readPacketTail** (**LtNetPacketTail** *tail)

Protected Attributes

- DtTime **myFirstTime**
- DtTime **myLastTime**
- long **myStartPos**
the ftell() on the start of the first packet. [More...](#)
- long **myEndPos**
the ftell() on the EOF. [More...](#)

Constructor & Destructor Documentation

LtLoggerPlaybackFile::LtLoggerPlaybackFile (const char * *FileName*)

Constructor.

LtLoggerPlaybackFile::~~LtLoggerPlaybackFile () [virtual]

Destructor.

Member Function Documentation

int LtLoggerPlaybackFile::atEnd () [virtual]

int LtLoggerPlaybackFile::atStart () [virtual]

Returns 1 if at beginning/end of file, 0 otherwise.

int LtLoggerPlaybackFile::checkMagicNumber () [protected, virtual]

Reads the magic number at beginning of file. Returns 1 if it matches the expected version, 0 otherwise.

int LtLoggerPlaybackFile::checkMagicNumber (const char * *filename*) [static]

This function checks filename's logger format Returns 1 if the file is valid; Prints error message and returns 0 otherwise.

int LtLoggerPlaybackFile::fileTimeBackwards (DtTime * *t*) [virtual]

Fills t with time of next or previous packet. Returns 1 on success, 0 if no packet in that direction.

int LtLoggerPlaybackFile::fileTimeForwards (DtTime * *t*) [virtual]

int LtLoggerPlaybackFile::firstPacketTime (DtTime * *t*) [virtual]

Fills t with time of first/last packet in file.

int LtLoggerPlaybackFile::jumpToEnd () [virtual]

int LtLoggerPlaybackFile::jumpToStart () [virtual]

Go to beginning/end of file.

```
int LtLoggerPlaybackFile::lastPacketTime (DtTime * t) [virtual]
```

```
LtPacketInfo * LtLoggerPlaybackFile::readPacket () [virtual]
```

Reads and returns next packet, NULL if none. In HLA, packets might be skipped if the file was created with a non-NULL exConn, and the packet cannot be converted to the exConn's FOM.

```
LtPacketInfo * LtLoggerPlaybackFile::readPacketBackwards () [virtual]
```

Reads and returns previous packet, NULL if none. In HLA, packets might be skipped if the file was created with a non-NULL exConn, and the packet cannot be converted to the exConn's FOM.

```
int LtLoggerPlaybackFile::readPacketData (void * buffer,  
                                           int bytesToRead) [protected, virtual]
```

```
int LtLoggerPlaybackFile::readPacketHeader (LtNetPacketHeader * header) [protected,  
                                     virtual]
```

Read header, data and tail into arguments. Return 1 if successful, 0 otherwise.

```
int LtLoggerPlaybackFile::readPacketTail (LtNetPacketTail * tail) [protected, virtual]
```

```
void LtLoggerPlaybackFile::seekBackward (DtTime t) [virtual]
```

```
void LtLoggerPlaybackFile::seekForward (DtTime t) [virtual]
```

Jump forward/backwards in file by t seconds.

void LtLoggerPlaybackFile::seekTime (DtTime *t*) [virtual]

Jump to time *t* (in seconds) in file.

int LtLoggerPlaybackFile::skipPacketBackwards () [virtual]

int LtLoggerPlaybackFile::skipPacketForwards () [virtual]

Skip over next packet either forwards or backwards direction. Return 1 if successful, 0 if no packets in that direction.

Member Data Documentation

long LtLoggerPlaybackFile::myEndPos [protected]

the ftell() on the EOF.

DtTime LtLoggerPlaybackFile::myFirstTime [protected]

DtTime LtLoggerPlaybackFile::myLastTime [protected]

long LtLoggerPlaybackFile::myStartPos [protected]

the ftell() on the start of the first packet.

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

- [loggerPbFile.h](#)

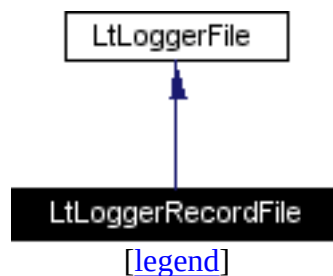
[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

LtLoggerRecordFile Class Reference

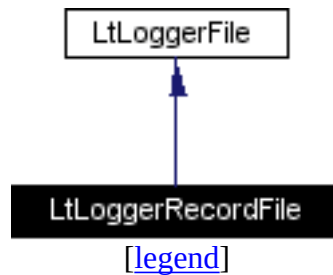
```
#include <loggerRecFile.h>
```

Inherits **LtLoggerFile**.

Inheritance diagram for LtLoggerRecordFile:



Collaboration diagram for LtLoggerRecordFile:



[List of all members.](#)

Public Methods

- **LtLoggerRecordFile** (const char *FileName)
Constructor. [More...](#)
- virtual ~**LtLoggerRecordFile** ()
Destructor. [More...](#)
- virtual int **writePacket** (**LtPacketInfo** *packetInfo)
Write a packet to the file. [More...](#)
- virtual int **writePacket** (void *buffer, int length, double time)

Write a packet to the file. [More...](#)

Protected Methods

- virtual int **writeMagicNumber** ()

Write version specific magic number to. [More...](#)

Constructor & Destructor Documentation

LtLoggerRecordFile::LtLoggerRecordFile (const char * *FileName*)

Constructor.

LtLoggerRecordFile::~~LtLoggerRecordFile () [virtual]

Destructor.

Member Function Documentation

int LtLoggerRecordFile::writeMagicNumber () [protected, virtual]

Write version specific magic number to.

int LtLoggerRecordFile::writePacket (void * *buffer*,
int *length*,
double *time*) [virtual]

Write a packet to the file.

int LtLoggerRecordFile::writePacket (**LtPacketInfo** * *packetInfo*) [virtual]

Write a packet to the file.

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

- [**loggerRecFile.h**](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

LtNetInteractionRecord Class Reference

```
#include <netPacket.h>
```

[List of all members.](#)

Public Attributes

- DtNetU32 [classHandle](#)
- DtNetU16 [numParams](#)
- DtNetU16 [numValueBytes](#)

Member Data Documentation

DtNetU32 LtNetInteractionRecord::classHandle

DtNetU16 LtNetInteractionRecord::numParams

DtNetU16 LtNetInteractionRecord::numValueBytes

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

- [netPacket.h](#)

LtNetPacketHeader Class Reference

```
#include <netPacket.h>
```

[List of all members.](#)

Public Attributes

- DtNetU32 **numBytes**
- DtNetU32 **padding**
- DtNetFloat64 **arrivalTime**

Member Data Documentation

DtNetFloat64 LtNetPacketHeader::arrivalTime

DtNetU32 LtNetPacketHeader::numBytes

DtNetU32 LtNetPacketHeader::padding

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

- **[netPacket.h](#)**

LtNetPacketTail Class Reference

```
#include <netPacket.h>
```

[List of all members.](#)

Public Attributes

- DtNetU32 **numBytes**

Member Data Documentation

DtNetU32 LtNetPacketTail::numBytes

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

- [netPacket.h](#)**

LtNetRemovalRecord Class Reference

```
#include <netPacket.h>
```

[List of all members.](#)

Public Attributes

- DtNetU32 **objectId**
- DtNetU32 **classHandle**
- DtNetU16 **nameSize**
- DtNetU16 **pad1**
- DtNetU32 **pad2**

Member Data Documentation

DtNetU32 LtNetRemovalRecord::classHandle

DtNetU16 LtNetRemovalRecord::nameSize

DtNetU32 LtNetRemovalRecord::objectId

DtNetU16 LtNetRemovalRecord::pad1

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

- [netPacket.h](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

LtNetUpdateRecord Class Reference

```
#include <netPacket.h>
```

[List of all members.](#)

Public Attributes

- DtNetU32 **objectId**
- DtNetU32 **classHandle**
- DtNetU16 **numAttrs**
- DtNetU16 **nameSize**
- DtNetU16 **numValueBytes**
size of name. [More...](#)
- DtNetU16 **padding**
size of values. [More...](#)

Member Data Documentation

DtNetU32 LtNetUpdateRecord::classHandle

DtNetU16 LtNetUpdateRecord::nameSize

DtNetU16 LtNetUpdateRecord::numAttrs

DtNetU16 LtNetUpdateRecord::numValueBytes

size of name.

DtNetU32 LtNetUpdateRecord::objectId

DtNetU16 LtNetUpdateRecord::padding

size of values.

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

- [**netPacket.h**](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

LtPacketInfo Class Reference

#include <packetInfo.h>

[List of all members.](#)

Public Methods

- **LtPacketInfo** ()
constructor. [More...](#)
- virtual **~LtPacketInfo** ()
destructor. [More...](#)
- **LtPacketInfo** (const LtPacketInfo &orig)
copy constructor. [More...](#)
- LtPacketInfo& **operator=** (const LtPacketInfo &orig)
assignment operator. [More...](#)
- virtual void **setPacketKind** (**LtPacketKind** theKind)
PacketKind. [More...](#)
- virtual const **LtPacketKind** **packetKind** () const
- virtual void **setArrivalTime** (DtTime theTime)
Arrival Time. [More...](#)
- virtual const DtTime **arrivalTime** () const
- virtual void **setSize** (int theSize)
DIS PDU size or size of a unknown HLA packet. [More...](#)
- virtual const int **size** () const
- virtual void **setPacket** (char *thePacket)
DIS PDU packet or unknown HLA buffer. [More...](#)
- virtual char* **packet** () const
- virtual int **writePacket** (char *buffer)

Write this packet to the buffer. returns the number of bytes written. [More...](#)

- virtual DtBoolean **readPacket** (const char *buffer)
read a buffer and set the values appropriately. [More...](#)

Protected Attributes

- **LtPacketKind myPacketKind**
- DtTime **myArrivalTime**
- int **mySize**
seconds. [More...](#)
- char* **myPacket**
bytes. [More...](#)

Constructor & Destructor Documentation

LtPacketInfo::LtPacketInfo ()

constructor.

LtPacketInfo::~~LtPacketInfo () [virtual]

destructor.

LtPacketInfo::LtPacketInfo (const LtPacketInfo & orig)

copy constructor.

Member Function Documentation

const DtTime LtPacketInfo::arrivalTime () const [virtual]

```
LtPacketInfo & LtPacketInfo::operator= (const LtPacketInfo & orig )
```

assignment operator.

```
char * LtPacketInfo::packet () const [virtual]
```

```
const LtPacketKind LtPacketInfo::packetKind () const [virtual]
```

```
DtBoolean LtPacketInfo::readPacket (const char * buffer ) [virtual]
```

read a buffer and set the values appropriately.

```
void LtPacketInfo::setArrivalTime (DtTime theTime ) [virtual]
```

Arrival Time.

```
void LtPacketInfo::setPacket (char * thePacket ) [virtual]
```

DIS PDU packet or unknown HLA buffer.

```
void LtPacketInfo::setPacketKind (LtPacketKind theKind ) [virtual]
```

PacketKind.

```
void LtPacketInfo::setSize (int theSize ) [virtual]
```

DIS PDU size or size of a unknown HLA packet.

```
const int LtPacketInfo::size () const [virtual]
```

```
int LtPacketInfo::writePacket (char * buffer ) [virtual]
```

Write this packet to the buffer. returns the number of bytes written.

Member Data Documentation

```
DtTime LtPacketInfo::myArrivalTime [protected]
```

```
char * LtPacketInfo::myPacket [protected]
```

bytes.

```
LtPacketKind LtPacketInfo::myPacketKind [protected]
```

```
int LtPacketInfo::mySize [protected]
```

seconds.

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

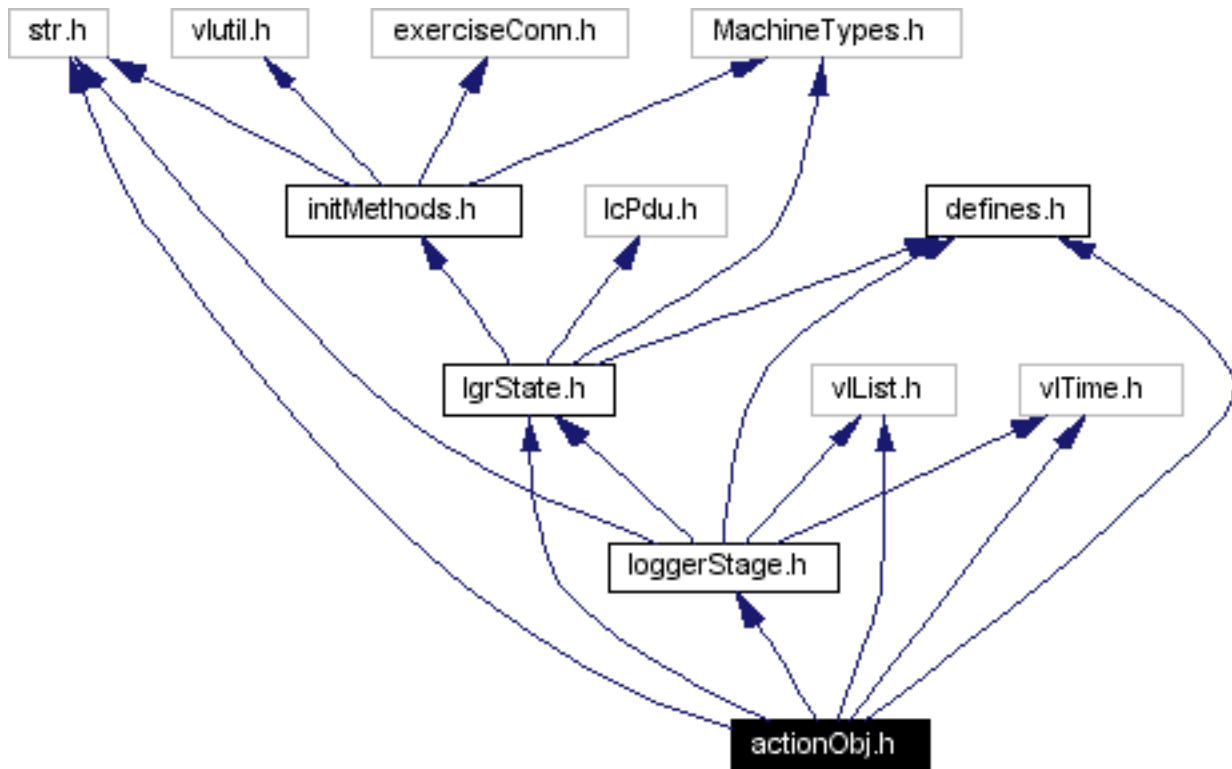
- [packetInfo.h](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

actionObj.h File Reference

```
#include "str.h"  
#include "vlList.h"  
#include "vlTime.h"  
#include "lgrState.h"  
#include "defines.h"  
#include "loggerStage.h"
```

Include dependency graph for actionObj.h:



Compounds

- class **DtActionObj**

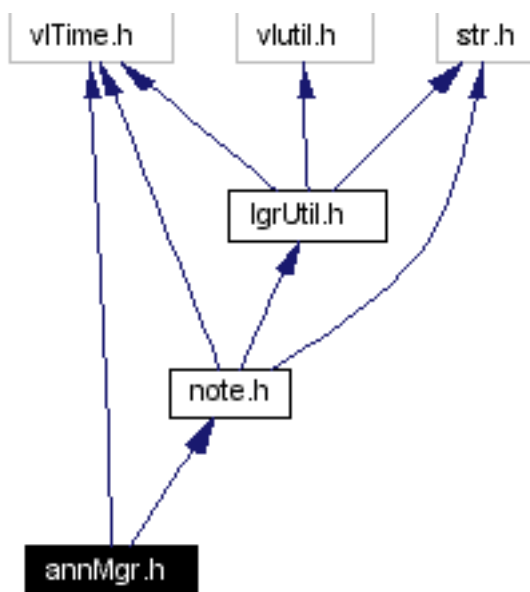
this is an abstract class that defines a action Object for the logger agent to perform specific actions. examples can be ramToools interface. [More...](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

annMgr.h File Reference

```
#include "note.h"  
#include "vlTime.h"
```

Include dependency graph for annMgr.h:

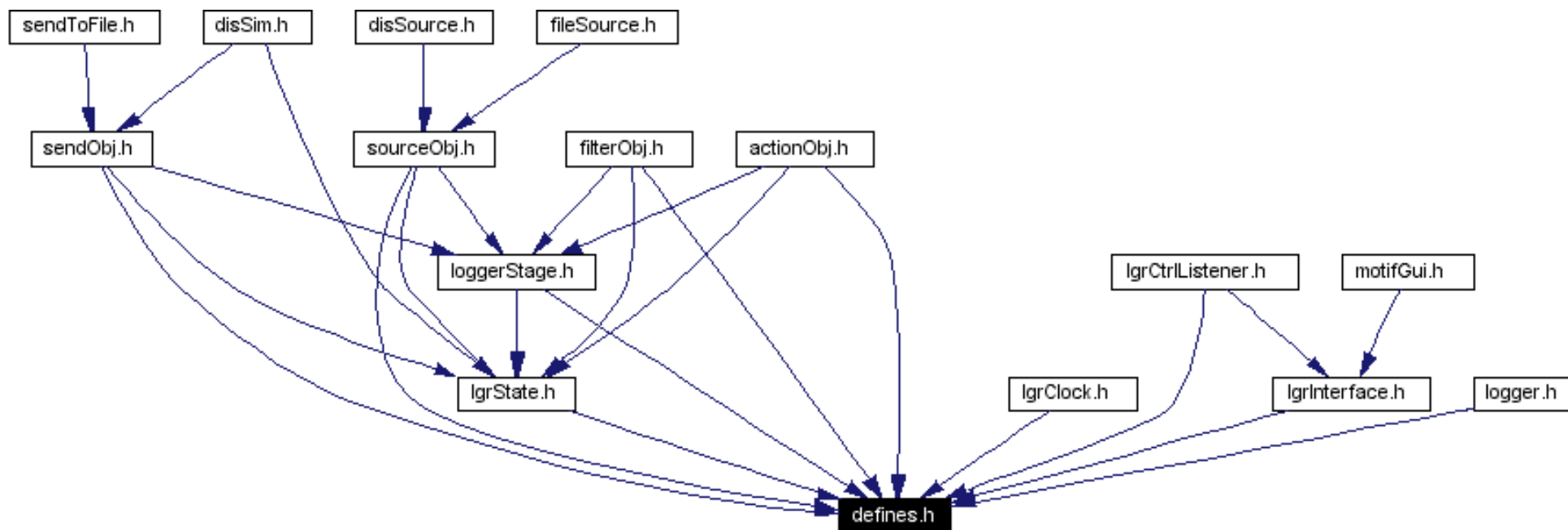


Compounds

- class [DtAnnotationManager](#)

defines.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define **AGENT_LIB**
!WIN32 no special qualififers are necessary on non-WIN32 platforms. [More...](#)
- #define **LGR_LIB**
- #define **LGR_APP**

Enumerations

- enum **LtObjectType** {
 LtUndefined, **LtSource**, **LtAction**, **LtFilter**,
 LtSend

}
Types of objects used in the Agent path. [More...](#)

Define Documentation

```
#define AGENT_LIB
```

```
!WIN32 no special qualifiers are necessary on non-WIN32 platforms.
```

```
#define LGR_APP
```

```
#define LGR_LIB
```

Enumeration Type Documentation

```
enum LtObjectType
```

```
Types of objects used in the Agent path.
```

Enumeration values:

LtUndefined

LtSource

LtAction

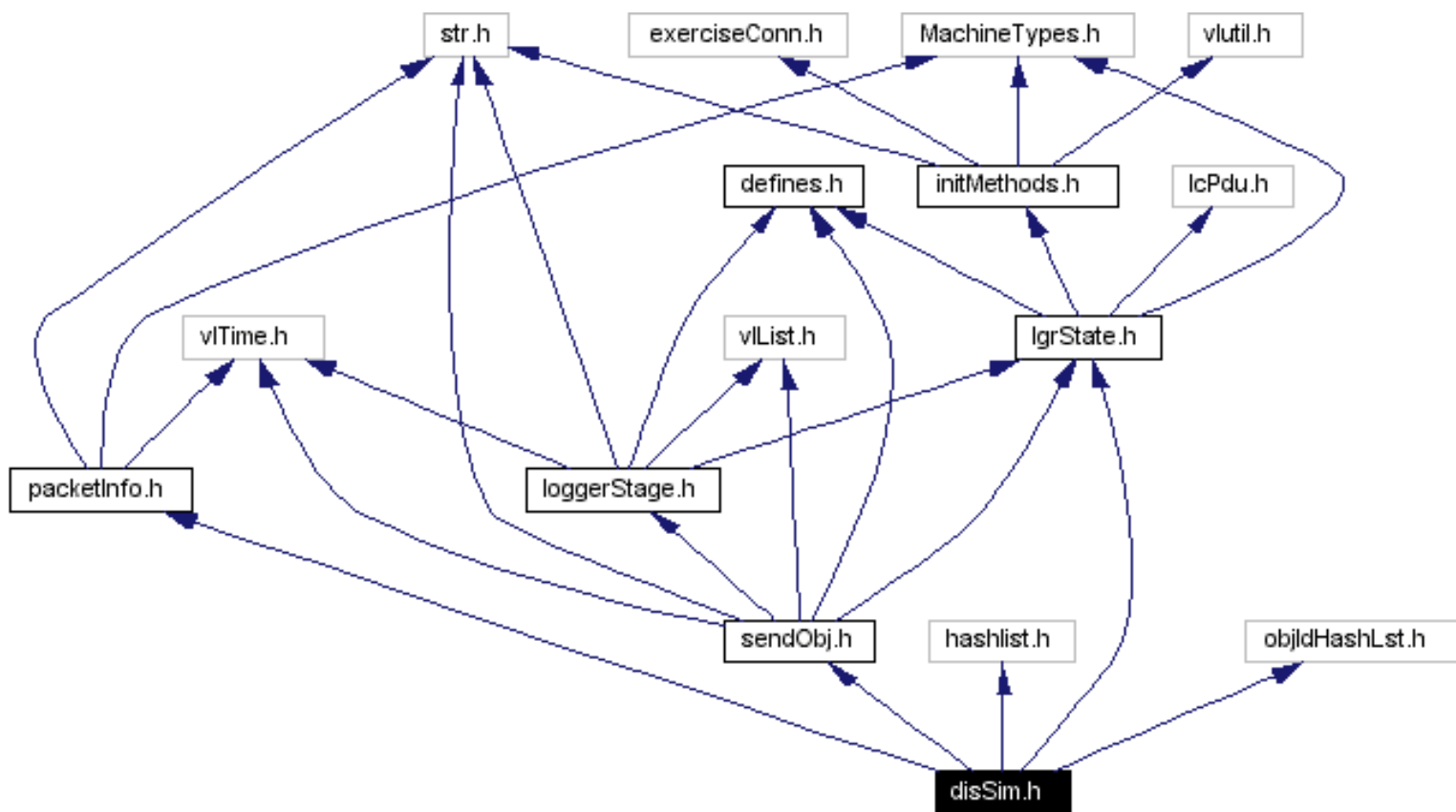
LtFilter

LtSend

disSim.h File Reference

```
#include "objIdHashLst.h"  
#include "sendObj.h"  
#include "hashlist.h"  
#include "packetInfo.h"  
#include "lgrState.h"
```

Include dependency graph for disSim.h:



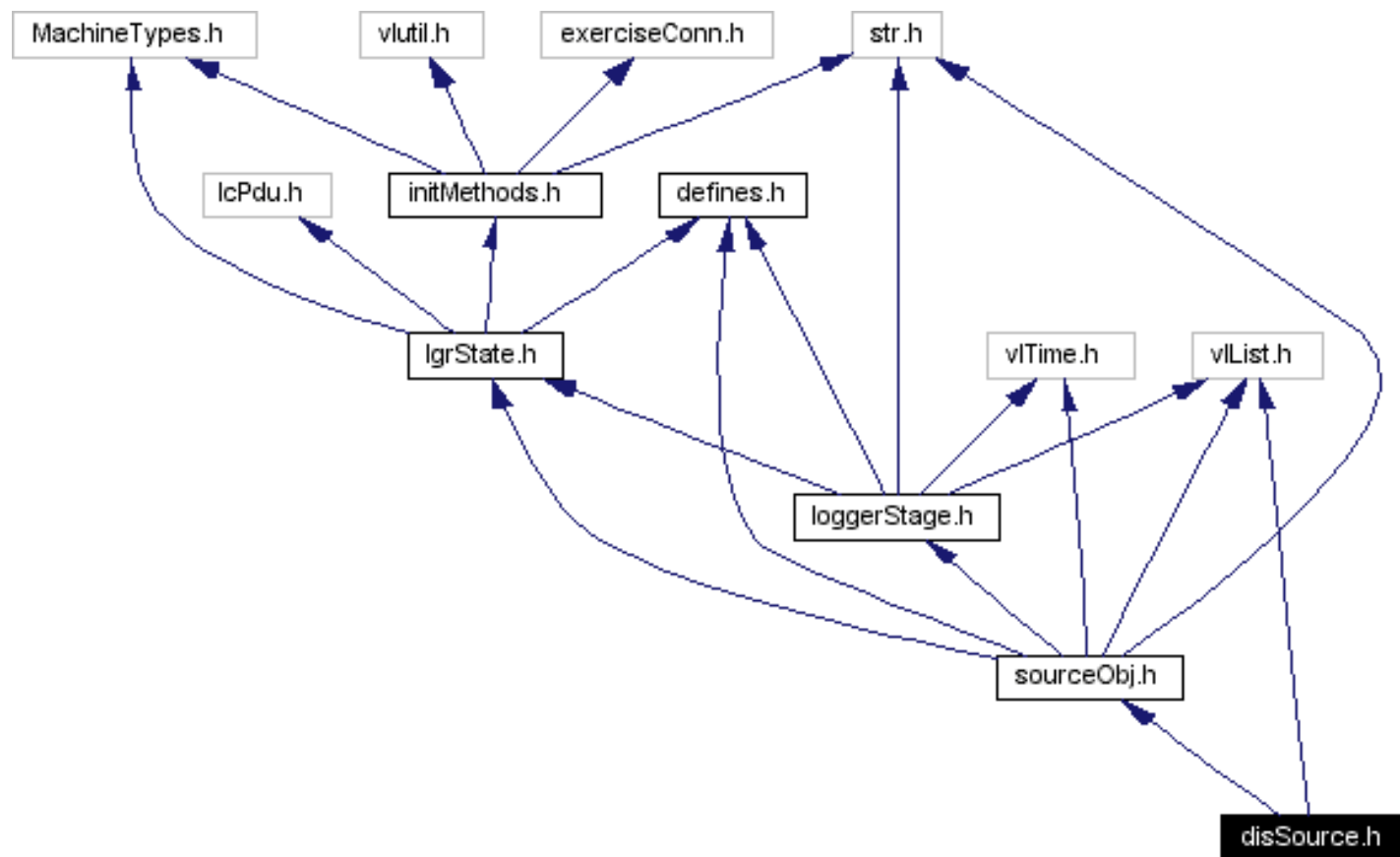
Compounds

- class [DtDisSimulate](#)

disSource.h File Reference

```
#include "sourceObj.h"
#include "vlList.h"
```

Include dependency graph for disSource.h:



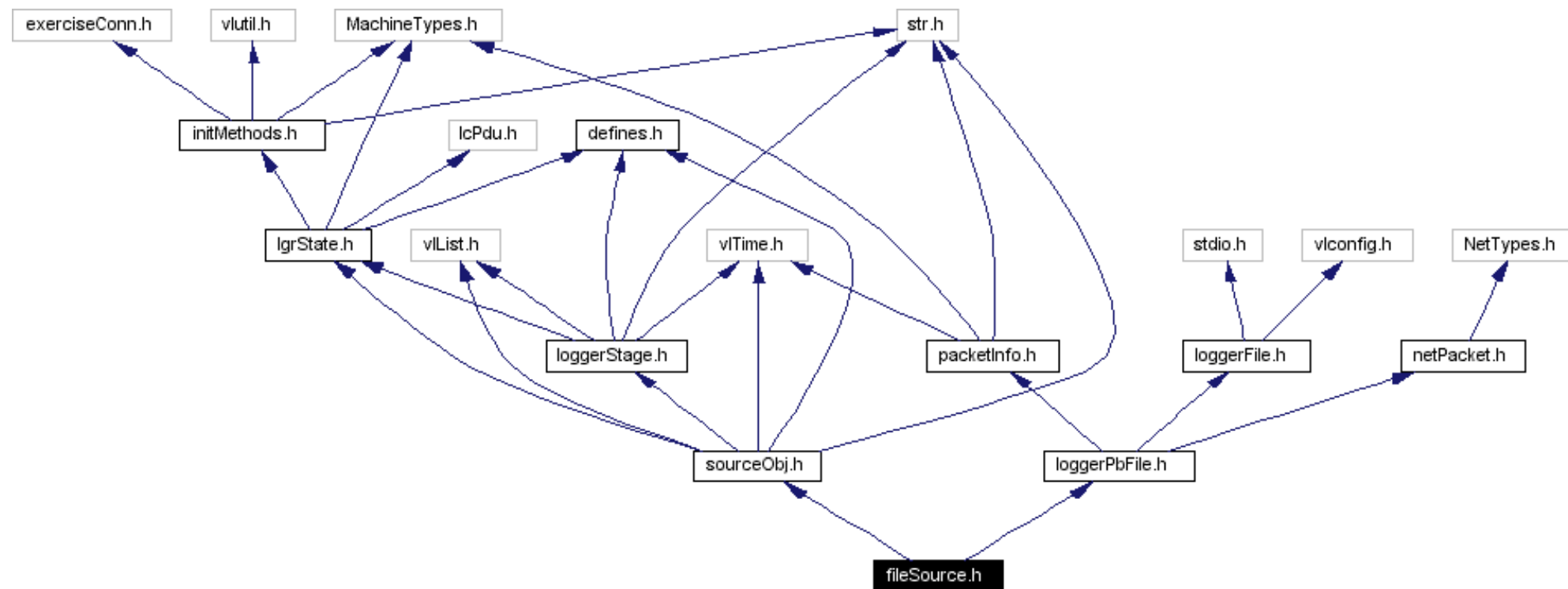
Compounds

- class **DtDisSource**

fileSource.h File Reference

```
#include "sourceObj.h"  
#include "loggerPbFile.h"
```

Include dependency graph for fileSource.h:



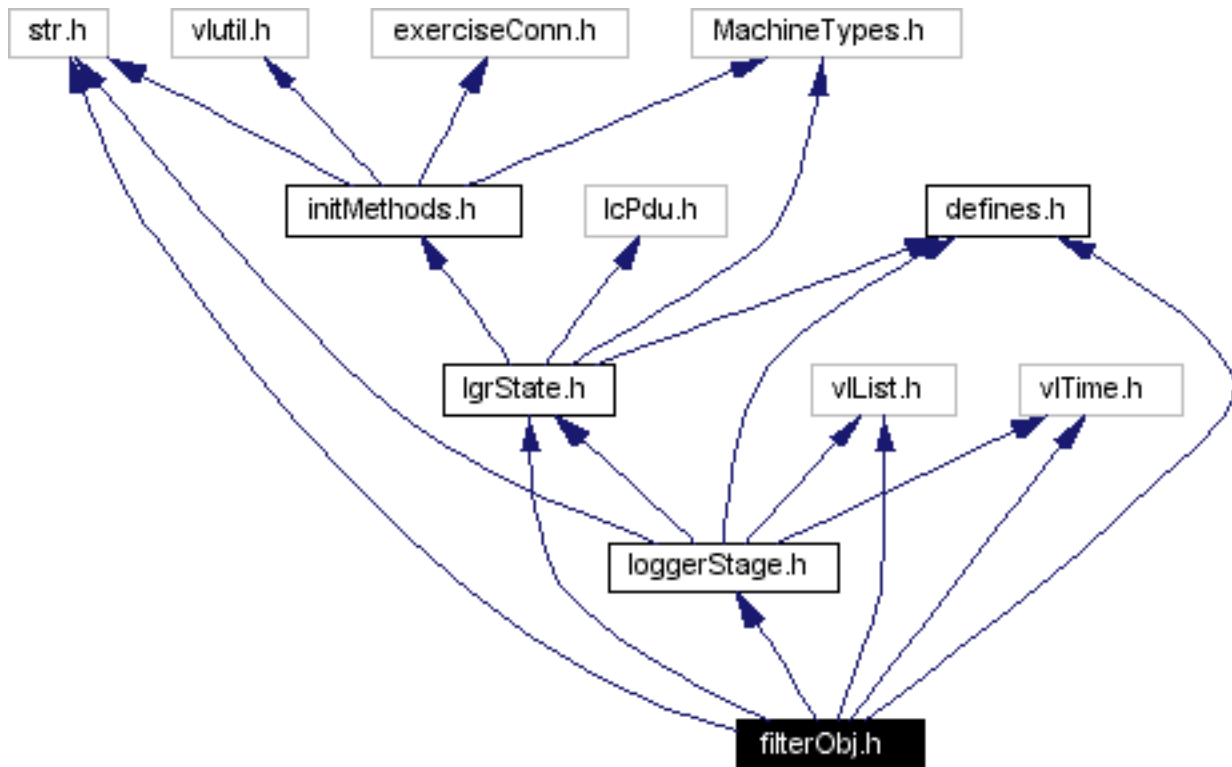
Compounds

- class [DtFileSource](#)

filterObj.h File Reference

```
#include "str.h"  
#include "vlList.h"  
#include "vlTime.h"  
#include "lgrState.h"  
#include "defines.h"  
#include "loggerStage.h"
```

Include dependency graph for filterObj.h:



Compounds

- class **DtFilterObj**

this is an abstract class that defines a filter for the logger agent to manage data. examples can be ramTooools interface. [More...](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

hActiveSource.h File Reference

hlaSim.h File Reference

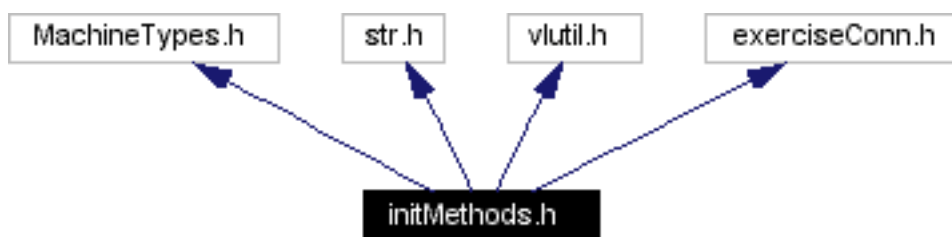
hlaSource.h File Reference

hPassSource.h File Reference

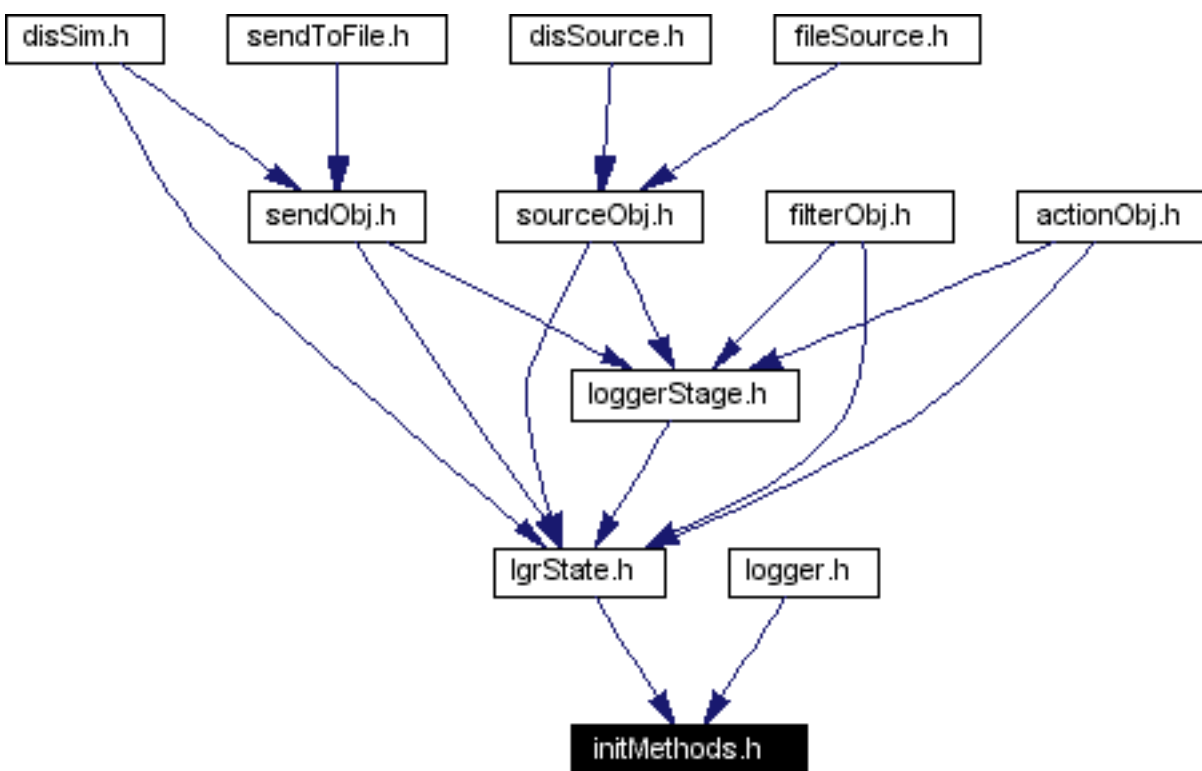
initMethods.h File Reference

```
#include "MachineTypes.h"  
#include "str.h"  
#include "vlutil.h"  
#include "exerciseConn.h"
```

Include dependency graph for initMethods.h:



This graph shows which files directly or indirectly include this file:



Defines

- #define **LGR_EXPORT**
windows export declaration to be used in initPlugin. [More...](#)

Typedefs

- typedef void* (* **DtCreateMethod**)(DtString)
This type defines a method used to create a instance of a lgr elt. [More...](#)
 - typedef **LtLoggerPlaybackFile*** (* **DtLoggerPlaybackConstructMethod**)(const char *filename)
These defines are to enable overloading of the loggerFile Playback and record classes. By overloading these classes the plugin writer can change the logger file format. [More...](#)
 - typedef **LtLoggerRecordFile*** (* **DtLoggerRecordConstructMethod**)(const char *filename)
-

Define Documentation

#define **LGR_EXPORT**

Value:

windows export declaration to be used in initPlugin.

Typedef Documentation

typedef void (* **DtCreateMethod**)(DtString)

This type defines a method used to create a instance of a lgr elt.

typedef **LtLoggerPlaybackFile** (* **DtLoggerPlaybackConstructMethod**)(const char *filename)

These defines are to enable overloading of the loggerFile Playback and record classes. By overloading these classes the plugin writer can change the logger file format.

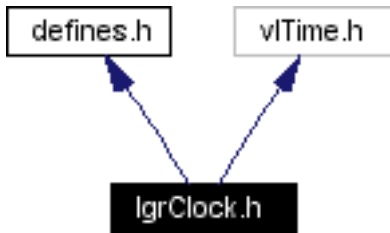
```
typedef LtLoggerRecordFile>(* DtLoggerRecordConstructMethod)(const char *filename)
```

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

IgrClock.h File Reference

```
#include "defines.h"  
#include "vlTime.h"
```

Include dependency graph for IgrClock.h:



Compounds

- class [DtLgrPbClock](#)
- class [DtLgrRecClock](#)
- class [DtLgrScalingClock](#)

Defines

- #define [LGR_CLOCK_FOREVER](#) 1e10
define forever to be 300 years away (roughly). [More...](#)
- #define [LGR_CLOCK_SMALL_SCALE](#) 1e-8
define a very small scale below which timeouts happen at forever. [More...](#)

Define Documentation

```
#define LGR_CLOCK_FOREVER 1e10
```

define forever to be 300 years away (roughly).

```
#define LGR_CLOCK_SMALL_SCALE 1e-8
```

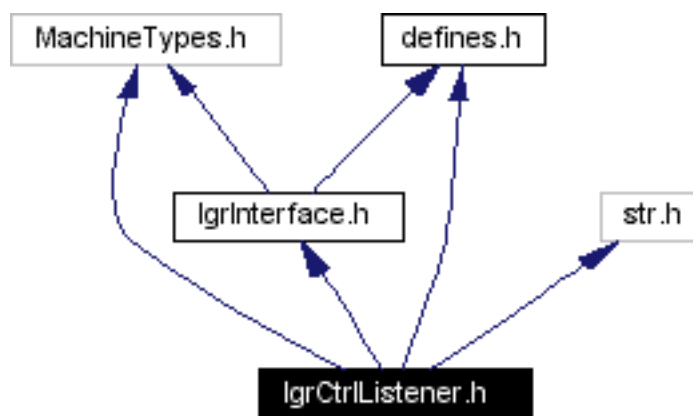
define a very small scale below which timeouts happen at forever.

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

lgrCtrlListener.h File Reference

```
#include "MachineTypes.h"
#include "str.h"
#include "defines.h"
#include "lgrInterface.h"
```

Include dependency graph for lgrCtrlListener.h:



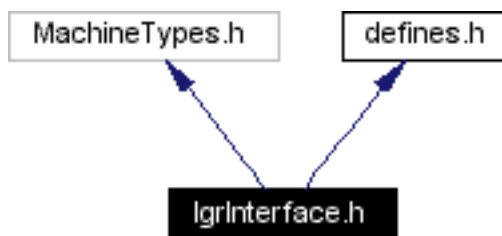
Compounds

- class [DtLgrControlListener](#)

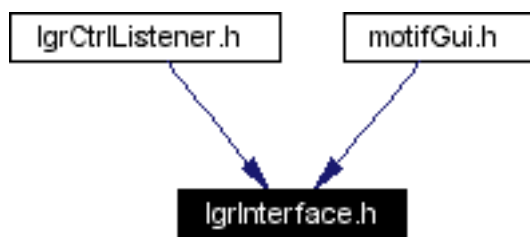
lgrInterface.h File Reference

```
#include "MachineTypes.h"  
#include "defines.h"
```

Include dependency graph for lgrInterface.h:



This graph shows which files directly or indirectly include this file:



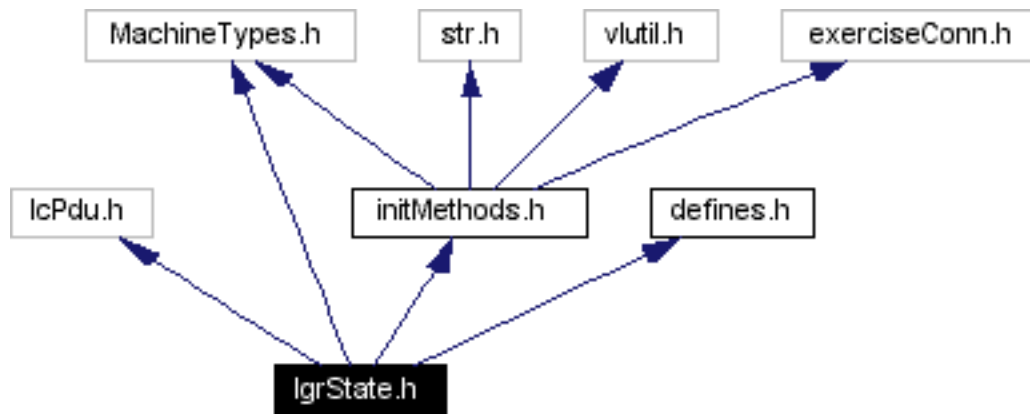
Compounds

- class [DtLgrInterface](#)

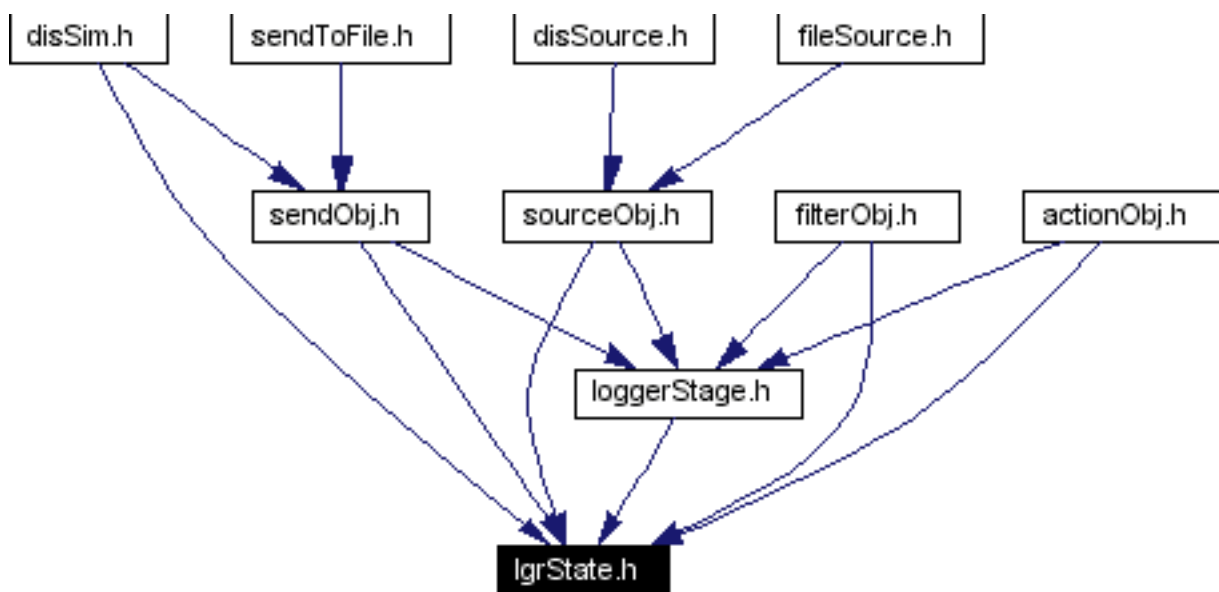
IgrState.h File Reference

```
#include "lcPdu.h"  
#include "MachineTypes.h"  
#include "initMethods.h"  
#include "defines.h"
```

Include dependency graph for IgrState.h:



This graph shows which files directly or indirectly include this file:



Compounds

- class [DtLgrState](#)

Defines

- #define [DEFAULT_PACKET_LIMIT](#) 1000

Define Documentation

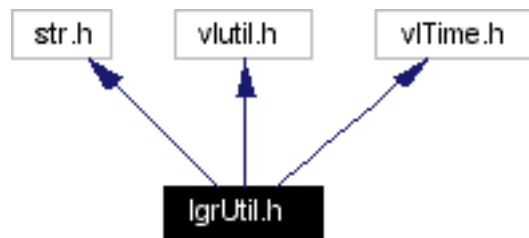
```
#define DEFAULT_PACKET_LIMIT 1000
```

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

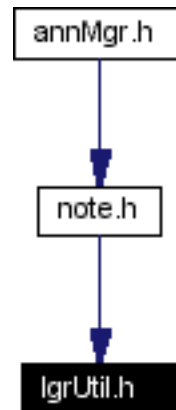
IgrUtil.h File Reference

```
#include "str.h"  
#include "vlutil.h"  
#include "vlTime.h"
```

Include dependency graph for IgrUtil.h:



This graph shows which files directly or indirectly include this file:



Functions

- char* **LtToLower** (char *inStr)
This is here temporarily, until we change vlutil so that DtToLower returns the value that it has modified.
[More...](#)
- DtBoolean **LtStrEquiv** (const char *one, const char *two)
This is here temporarily until it goes into DtString. Returns true if the strings match, disregarding case, else false. [More...](#)

- char* **LtChopSlash** (char *f)
 - void **LtTimeToString** (DtTime time, char *string)
 - DtTime **LtStringToTime** (char *timeString, int *errorFlag)
-

Function Documentation

```
char * LtChopSlash (char * f)
```

```
DtBoolean LtStrEquiv (const char * one,  
                     const char * two ) [inline]
```

This is here temporarily until it goes into DtString. Returns true if the strings match, disregarding case, else false.

```
DtTime LtStringToTime (char * timeString,  
                      int * errorFlag )
```

```
void LtTimeToString (DtTime time,  
                    char * string )
```

```
char * LtToLower (char * inStr ) [inline]
```

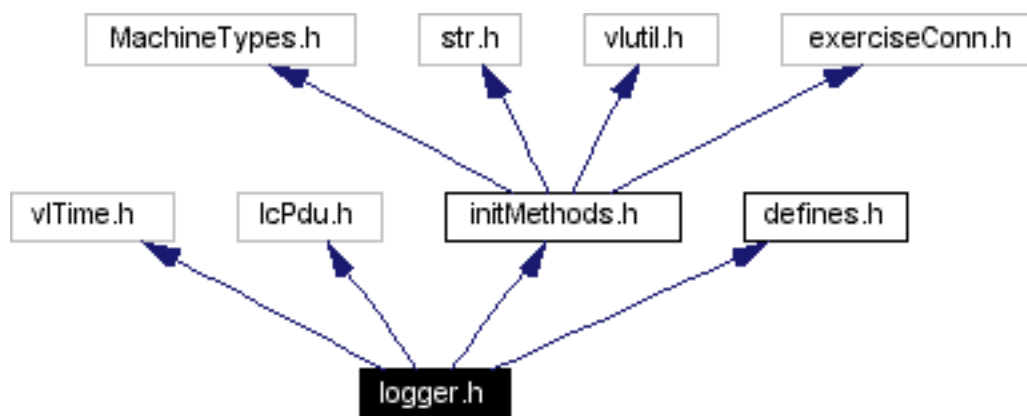
This is here temporarily, until we change vlutil so that DtToLower returns the value that it has modified.

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

logger.h File Reference

```
#include "vlTime.h"  
#include "lcPdu.h"  
#include "initMethods.h"  
#include "defines.h"
```

Include dependency graph for logger.h:



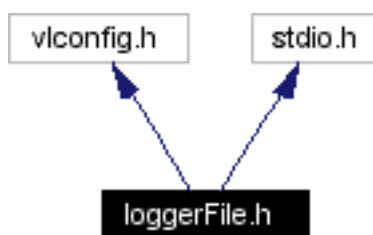
Compounds

- class [DtLogger](#)

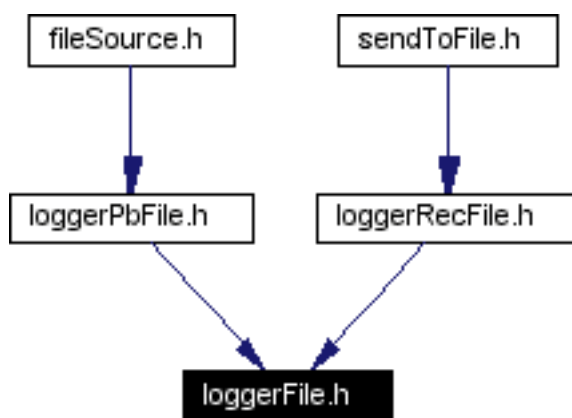
loggerFile.h File Reference

```
#include "vlconfig.h"  
#include <stdio.h>
```

Include dependency graph for loggerFile.h:



This graph shows which files directly or indirectly include this file:



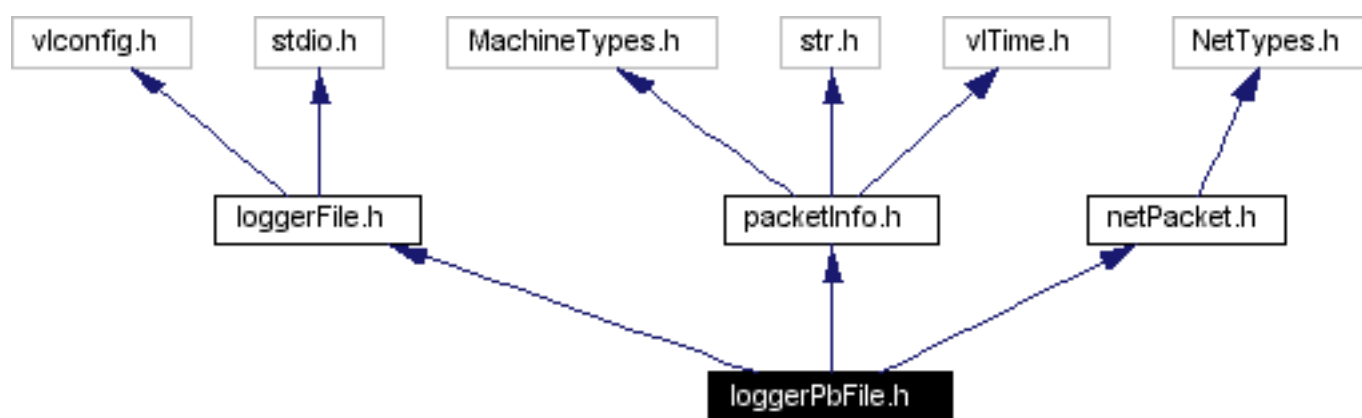
Compounds

- class **LtLoggerFile**
For DT_HAVE_UNISTD_H. [More...](#)

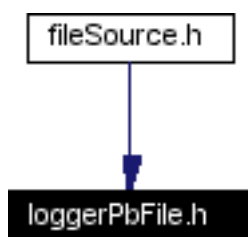
loggerPbFile.h File Reference

```
#include "loggerFile.h"  
#include "packetInfo.h"  
#include "netPacket.h"
```

Include dependency graph for loggerPbFile.h:



This graph shows which files directly or indirectly include this file:



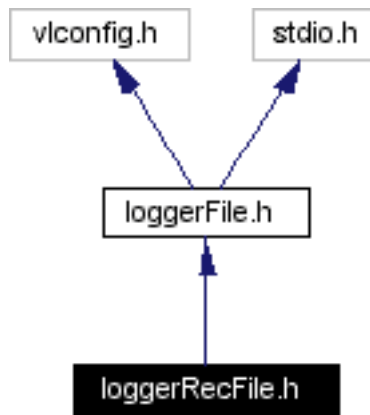
Compounds

- class [LtLoggerPlaybackFile](#)

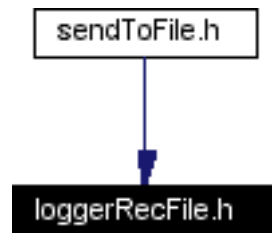
loggerRecFile.h File Reference

```
#include "loggerFile.h"
```

Include dependency graph for loggerRecFile.h:



This graph shows which files directly or indirectly include this file:



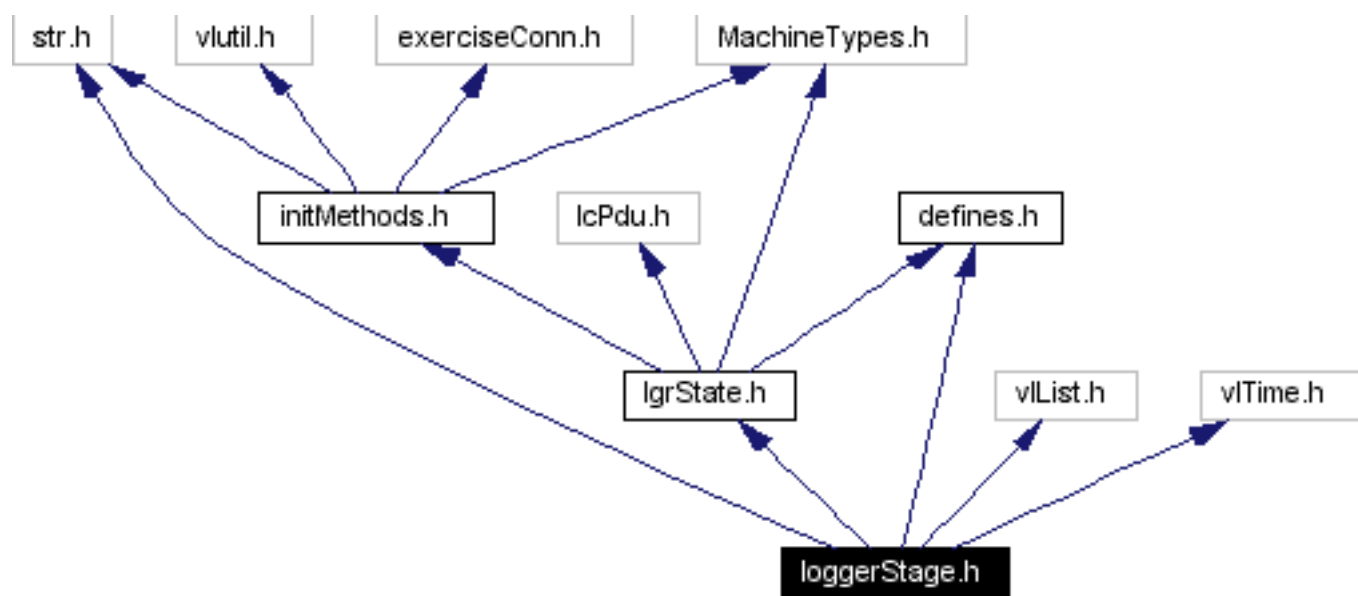
Compounds

- class [LtLoggerRecordFile](#)

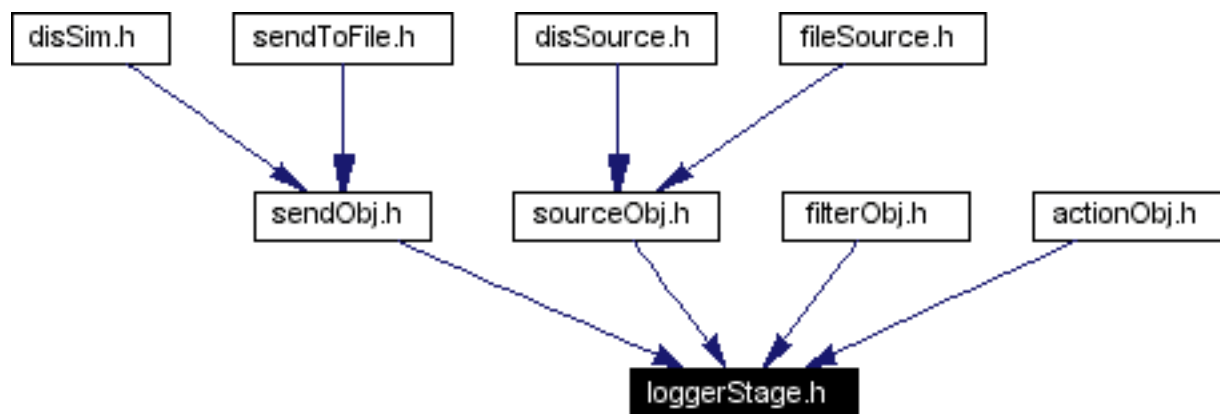
loggerStage.h File Reference

```
#include "str.h"  
#include "vlList.h"  
#include "vlTime.h"  
#include "lgrState.h"  
#include "defines.h"
```

Include dependency graph for loggerStage.h:



This graph shows which files directly or indirectly include this file:



Compounds

- class **DtLoggerStage**

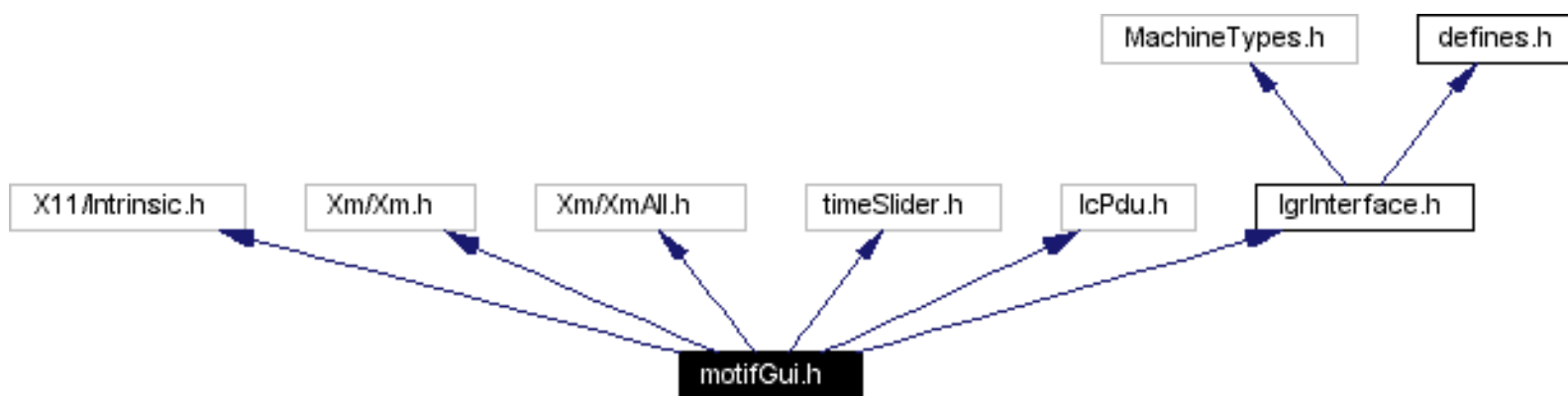
*this is an abstract class that defines a logger pipeline Object for the logger agent to perform specific actions.
examples can be ramToools interface. [More...](#)*

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

motifGui.h File Reference

```
#include <X11/Intrinsic.h>
#include <Xm/Xm.h>
#include <Xm/XmAll.h>
#include "timeSlider.h"
#include "lcPdu.h"
#include "lgrInterface.h"
```

Include dependency graph for motifGui.h:



Compounds

- class [DtMotifGui](#)

Defines

- #define [MAX_ANNOTS](#) 500

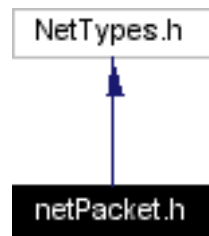
Define Documentation

```
#define MAX_ANNOTS 500
```

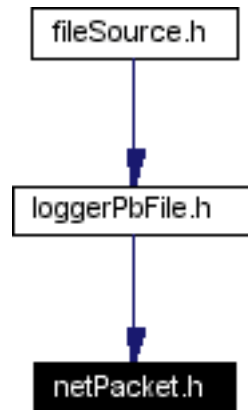

netPacket.h File Reference

```
#include "NetTypes.h"
```

Include dependency graph for netPacket.h:



This graph shows which files directly or indirectly include this file:



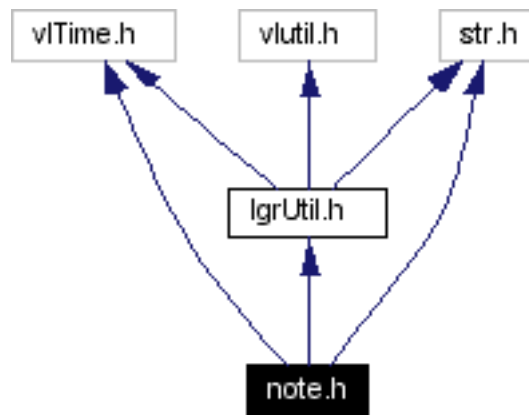
Compounds

- class [LtNetInteractionRecord](#)
- class [LtNetPacketHeader](#)
- class [LtNetPacketTail](#)
- class [LtNetRemovalRecord](#)
- class [LtNetUpdateRecord](#)

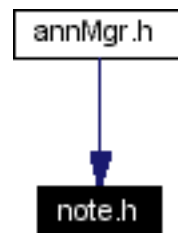
note.h File Reference

```
#include "vlTime.h"  
#include "str.h"  
#include "lgrUtil.h"
```

Include dependency graph for note.h:



This graph shows which files directly or indirectly include this file:



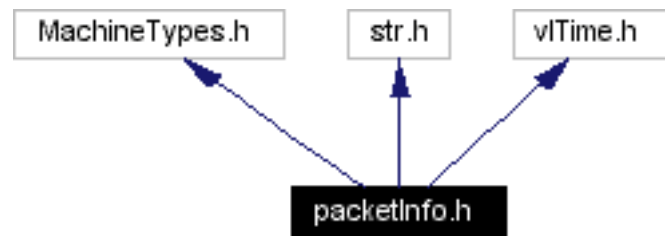
Compounds

- class [DtNote](#)

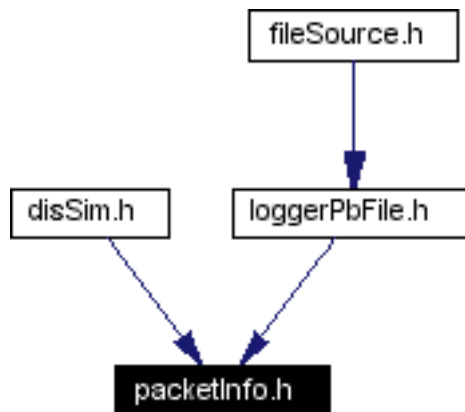
packetInfo.h File Reference

```
#include "MachineTypes.h"  
#include "str.h"  
#include "vlTime.h"
```

Include dependency graph for packetInfo.h:



This graph shows which files directly or indirectly include this file:



Compounds

- class [LtPacketInfo](#)

Defines

- #define [LtMAX_PACKET](#) 4440

Enumerations

- enum **LtPacketKind** {
 LtPacketInteraction = 0, **LtPacketUpdate**, **LtPacketRemoval**, **LtPacketAnnotation**,
 LtPacketPdu, **LtPacketFom**, **LtPacketAudio**
}
-

Define Documentation

```
#define LtMAX_PACKET 4440
```

Enumeration Type Documentation

enum LtPacketKind

Enumeration values:

LtPacketInteraction

LtPacketUpdate

LtPacketRemoval

LtPacketAnnotation

LtPacketPdu

LtPacketFom

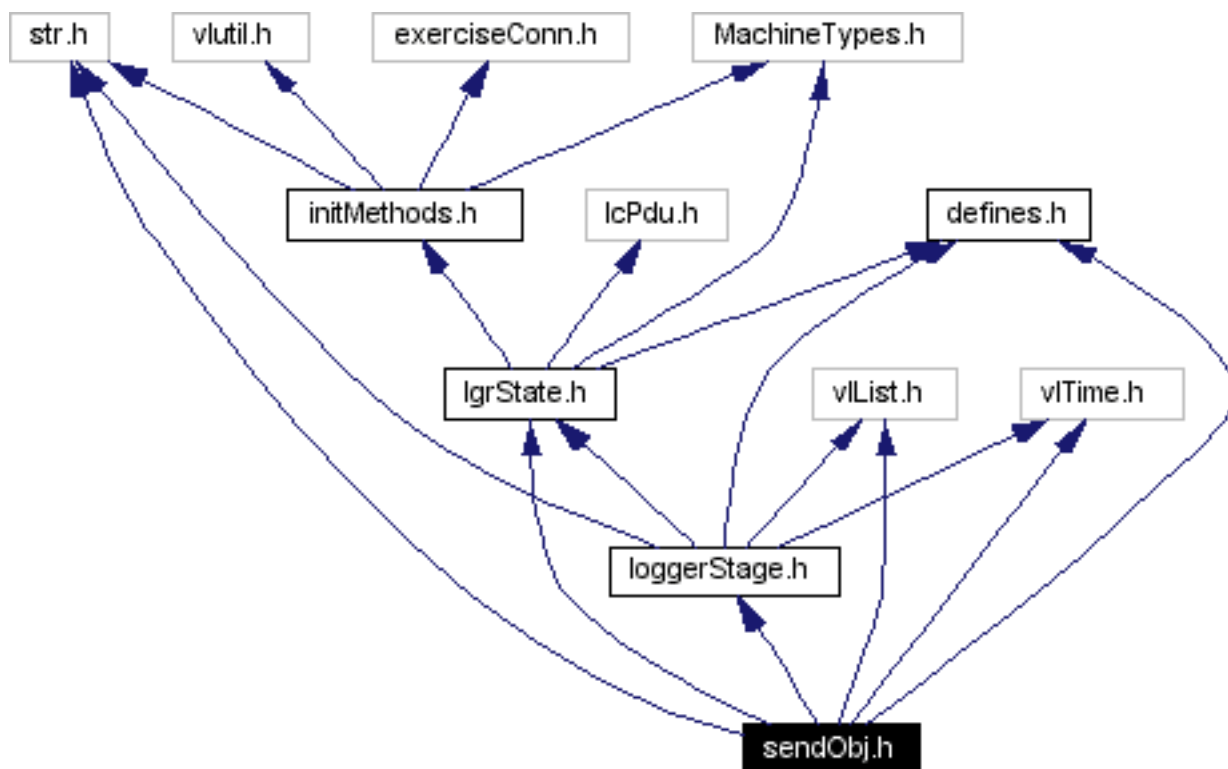
LtPacketAudio

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

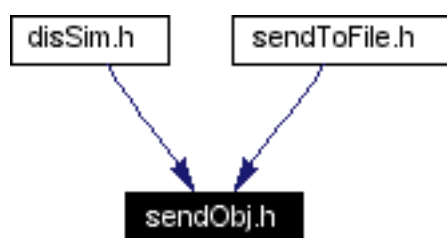
sendObj.h File Reference

```
#include "str.h"  
#include "vlList.h"  
#include "vlTime.h"  
#include "lgrState.h"  
#include "defines.h"  
#include "loggerStage.h"
```

Include dependency graph for sendObj.h:



This graph shows which files directly or indirectly include this file:



Compounds

- class **DtSendObj**

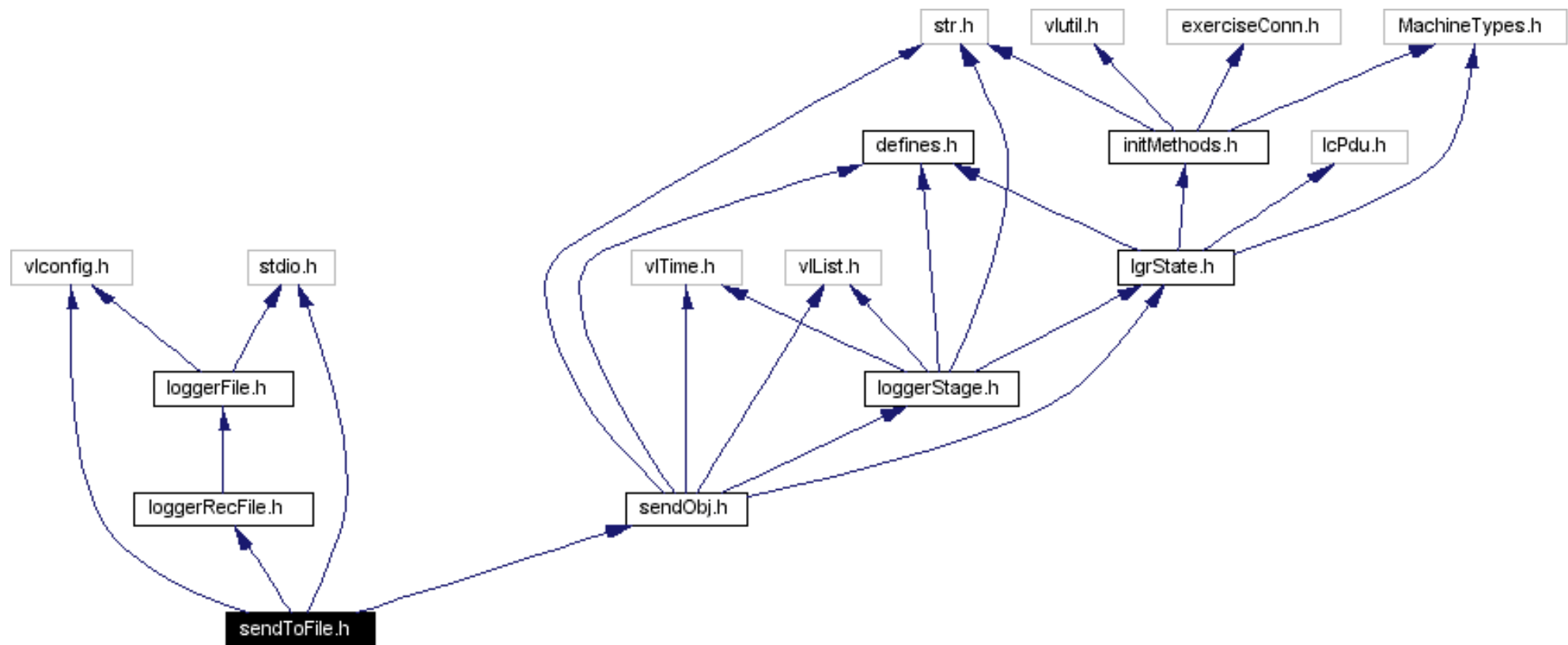
this is an abstract class that defines a target for the logger agent to send data. examples can be files, databases or a hla federation. [More...](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

sendToFile.h File Reference

```
#include "vlconfig.h"  
#include <stdio.h>  
#include "sendObj.h"  
#include "loggerRecFile.h"
```

Include dependency graph for sendToFile.h:



Compounds

- class [DtSendToFile](#)

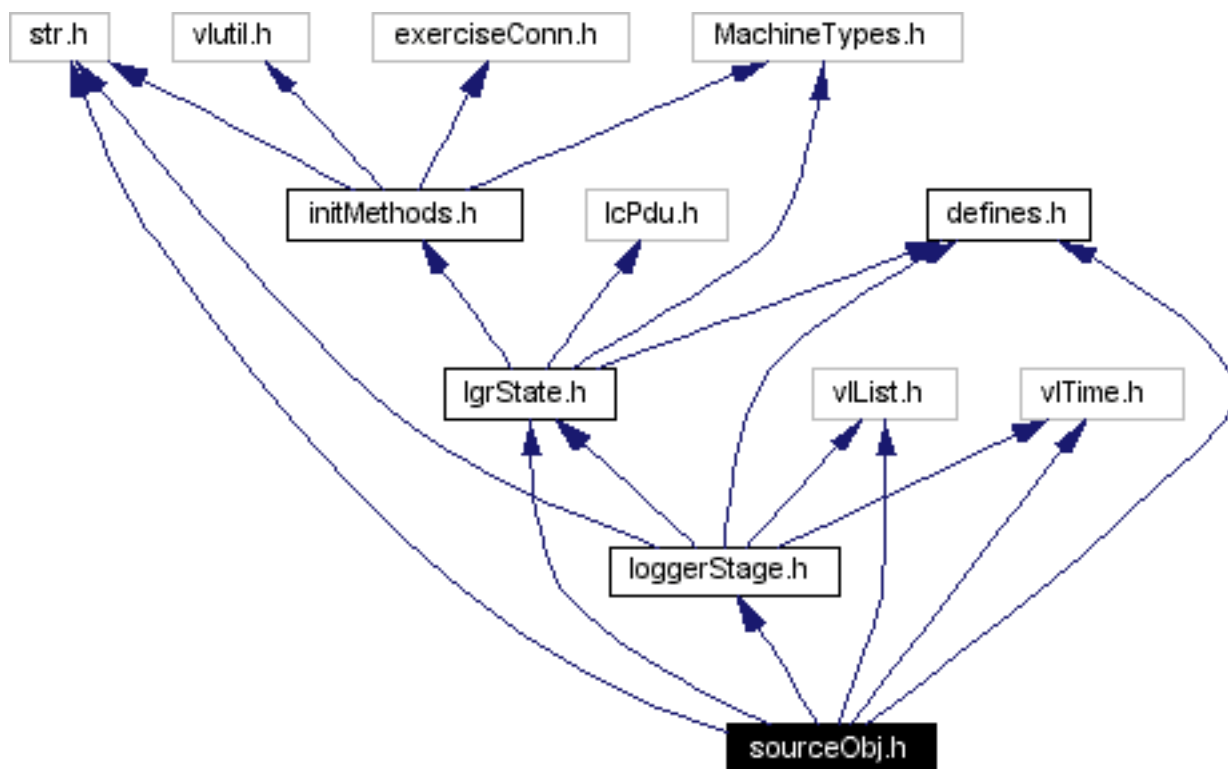
For `DT_HAVE_UNISTD_H` this class will send a packet to a file. [More...](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#) [File Members](#)

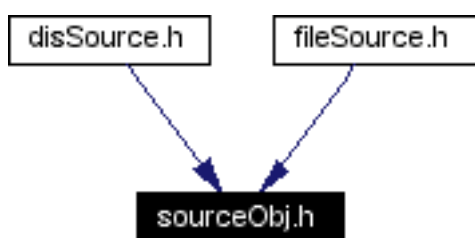
sourceObj.h File Reference

```
#include "str.h"  
#include "vlList.h"  
#include "vlTime.h"  
#include "lgrState.h"  
#include "defines.h"  
#include "loggerStage.h"
```

Include dependency graph for sourceObj.h:



This graph shows which files directly or indirectly include this file:



Compounds

- class **DtSourceObj**

this is an abstract class that defines a data source for a logger agent. A source could be a HLA connection, TCP connection, etc. Each connection will override these virtual functions and produce conforming data to enter the data stream. [More...](#)

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

stateInfo.h File Reference

winGui.h File Reference

Graph Legend

This page explains how to interpret the graphs that are generated by doxygen.

Consider the following example:

```
/*! Invisible class because of truncation */
class Invisible { };

/*! Truncated class, inheritance relation is hidden */
class Truncated : public Invisible { };

/* Class not documented with doxygen comments */
class Undocumented { };

/*! Class that is inherited using public inheritance */
class PublicBase : public Truncated { };

/*! Class that is inherited using protected inheritance */
class ProtectedBase { };

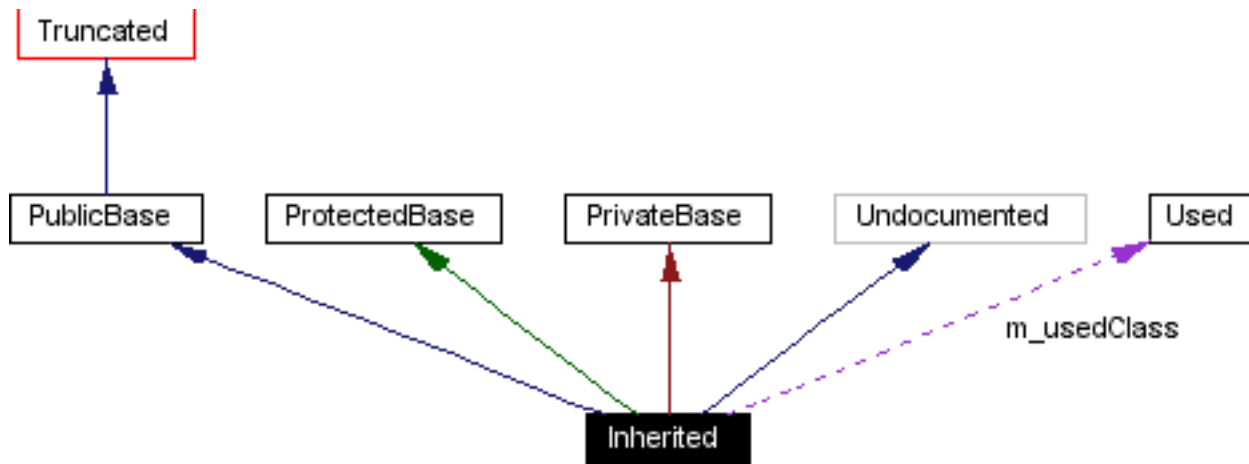
/*! Class that is inherited using private inheritance */
class PrivateBase { };

/*! Class that is used by the Inherited class */
class Used { };

/*! Super class that inherits a number of other classes */
class Inherited : public PublicBase,
                  protected ProtectedBase,
                  private PrivateBase,
                  public Undocumented
{
    private:
        Used *m_usedClass;
};
```

If the MAX_DOT_GRAPH_HEIGHT tag in the configuration file is set to 200 this will result in the

following graph:



The boxes in the above graph have the following meaning:

- A filled black box represents the struct or class for which the graph is generated.
- A box with a black border denotes a documented struct or class.
- A box with a grey border denotes an undocumented struct or class.
- A box with a red border denotes a documented struct or class for which not all inheritance/containment relations are shown. A graph is truncated if it does not fit within the specified boundaries.

The arrows have the following meaning:

- A dark blue arrow is used to visualize a public inheritance relation between two classes.
- A dark green arrow is used for protected inheritance.
- A dark red arrow is used for private inheritance.
- A purple dashed arrow is used if a class is contained or used by another class. The arrow is labeled with the variable(s) through which the pointed class or struct is accessible.

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtActionObj Member List

This is the complete list of members for [DtActionObj](#), including all inherited members.

- [activate\(\)](#)=0 [pure virtual]
- [actOnPacket](#)(const LtPacketInfo *packet)=0 [pure virtual]
- [deactivate\(\)](#)=0 [pure virtual]
- [DtActionObj](#)(const char *theName)
- [DtLoggerStage](#)(const char *theName)
- [DtLoggerStage](#)(const DtLoggerStage &) [protected]
- [getName](#)() const [inline, virtual]
- [initSelf](#)(DtLgrState *theState)=0 [pure virtual]
- [isOk](#)() const [inline, virtual]
- [myName](#) [protected]
- [myStatus](#) [protected]
- [printInfo](#)() const=0 [pure virtual]
- [processData](#)(DtList *packetList) [virtual]
- [reset\(\)](#)=0 [pure virtual]
- [tick\(\)](#)=0 [pure virtual]
- [~DtActionObj](#)() [virtual]
- [~DtLoggerStage](#)() [virtual]

DtAnnotationManager Member List

This is the complete list of members for [DtAnnotationManager](#), including all inherited members.

- [add](#)(DtTime t, DtString name) [virtual]
- [count](#)() const [virtual]
- [DtAnnotationManager](#)(int maxAnnotations=100)
- [empty](#)() [virtual]
- [first](#)() [virtual]
- [get](#)(DtTime t) [virtual]
- [loadFromFile](#)(const char *filename) [virtual]
- [myAnnotations](#) [protected]
- [myCurrent](#) [protected]
- [myLast](#) [protected]
- [myMax](#) [protected]
- [next](#)() [virtual]
- [remove](#)(DtTime t) [virtual]
- [saveToFile](#)(const char *filename) [virtual]
- [~DtAnnotationManager](#)() [virtual]

DtDisSimulate Member List

This is the complete list of members for **DtDisSimulate**, including all inherited members.

- **activate()** [virtual]
- **create**(DtString param) [static]
- **deactivate()** [virtual]
- **deleteAllObjs()** [protected, virtual]
- **DtDisSimulate**(DtString theName)
- **DtLoggerStage**(const char *theName)
- **DtLoggerStage**(const DtLoggerStage &) [protected]
- **DtSendObj**(const char *theName)
- **forceUpdatesOnAll()** [protected, virtual]
- **getName()** const [virtual]
- **initSelf**(DtLgrState *theState) [virtual]
- **isOk()** const [virtual]
- **myName** [protected]
- **myStatus** [protected]
- **printInfo()** const [virtual]
- **procAggPdu**(char *buff, int, DtBoolean sendFlag) [protected]
- **procEsPdu**(char *buff, int, DtBoolean sendFlag) [protected]
- **processData**(const DtList *packetList) [virtual]
- **reset()** [inline, virtual]
- **send**(LtPacketInfo *info) [protected, virtual]
- **sendPacket**(const LtPacketInfo *packet) [inline, virtual]
- **tick()** [protected, virtual]
- **~DtDisSimulate()**
- **~DtLoggerStage()** [virtual]
- **~DtSendObj()** [virtual]

DtDisSource Member List

This is the complete list of members for **DtDisSource**, including all inherited members.

- **activate**(DtTime startTime) [virtual]
- **activate**() [inline, virtual]
- **create**(DtString param) [static]
- **deactivate**() [virtual]
- **DtDisSource**(DtString theName)
- **DtLoggerStage**(const char *theName)
- **DtLoggerStage**(const DtLoggerStage &) [protected]
- **DtSourceObj**(const char *theName)
- **generatePacket**(DtTime currentTime, DtTime *nextTime) [inline, virtual]
- **generatePackets**(DtTime currentTime, DtList *packetList) [virtual]
- **getName**() const [virtual]
- **initSelf**(DtLgrState *theAIF) [virtual]
- **isOk**() const [virtual]
- **myActive** [protected]
- **myListenFlag** [protected]
- **myName** [protected]
- **myPacketList** [protected]
- **myState** [protected]
- **myStatus** [protected]
- **printInfo**() const [virtual]
- **recClock** [protected]
- **receivePdu**(DtPdu *pdu) [virtual]
- **receivePduCb**(DtPdu *pdu, void *usr) [static]
- **reset**() [virtual]
- **startListening**() [virtual]
- **tick**() [inline, virtual]
- **~DtDisSource**() [virtual]
- **~DtLoggerStage**() [virtual]
- **~DtSourceObj**() [virtual]

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtFileSource Member List

This is the complete list of members for **DtFileSource**, including all inherited members.

- **activate**(DtTime startTime) [virtual]
- **activate**() [inline, virtual]
- **create**(DtString param) [static]
- **deactivate**() [virtual]
- **DtFileSource**(DtString theName)
- **DtLoggerStage**(const char *theName)
- **DtLoggerStage**(const DtLoggerStage &) [protected]
- **DtSourceObj**(const char *theName)
- **generatePacket**(DtTime currentTime, DtTime *nextTime) [inline, virtual]
- **generatePackets**(DtTime currentTime, DtList *packetList) [virtual]
- **getName**() const [virtual]
- **initSelf**(DtLgrState *theState) [virtual]
- **isOk**() const [virtual]
- **myName** [protected]
- **myStatus** [protected]
- **printInfo**() const [virtual]
- **reset**() [virtual]
- **tick**() [virtual]
- **~DtFileSource**() [virtual]
- **~DtLoggerStage**() [virtual]
- **~DtSourceObj**() [virtual]

DtFilterObj Member List

This is the complete list of members for **DtFilterObj**, including all inherited members.

- **activate()**=0 [pure virtual]
- **deactivate()**=0 [pure virtual]
- **DtFilterObj**(const char *theName)
- **DtLoggerStage**(const char *theName)
- **DtLoggerStage**(const DtLoggerStage &) [protected]
- **filterPacket**(LtPacketInfo *packet)=0 [pure virtual]
- **getName()** const [inline, virtual]
- **initSelf**(DtLgrState *theState)=0 [pure virtual]
- **isOk()** const [inline, virtual]
- **myName** [protected]
- **myStatus** [protected]
- **printInfo()** const=0 [pure virtual]
- **processData**(DtList *packetList) [virtual]
- **reset()**=0 [pure virtual]
- **tick()** [inline, virtual]
- **~DtFilterObj()** [virtual]
- **~DtLoggerStage()** [virtual]

DtLgrControlListener Member List

This is the complete list of members for **DtLgrControlListener**, including all inherited members.

- **allowingFileDeletion**() const [virtual]
- **DtLgrControlListener**(DtLogger *logger, DtString id, DtBoolean processAll=DtTrue)
- **DtLgrInterface**(DtLogger *logger) [inline]
- **id**() const [virtual]
- **lgrControlCb**(DtInteraction *lcInter, void *theAction) [static]
- **myAllowFileDelete**
- **myId**
- **myLogger**
- **myProcessAll**
- **processAll**() const [virtual]
- **processLgrControl**(void *usr) [protected, static]
- **setAllowFileDeletion**(DtBoolean yesNo) [virtual]
- **setId**(const DtString &id) [virtual]
- **setProcessAll**(DtBoolean yesNo) [virtual]
- **setTimeToQuit**(DtBoolean yesNo) [inline, virtual]
- **tick**() [inline, virtual]
- **timeToQuit**() [inline, virtual]
- **~DtLgrControlListener**() [virtual]
- **~DtLgrInterface**() [inline, virtual]

DtLgrInterface Member List

This is the complete list of members for **DtLgrInterface**, including all inherited members.

- **DtLgrInterface**(DtLogger *logger) [inline]
- **myLogger**
- **setTimeToQuit**(DtBoolean yesNo) [inline, virtual]
- **tick**() [inline, virtual]
- **timeToQuit**() [inline, virtual]
- **~DtLgrInterface**() [inline, virtual]

DtLgrPbClock Member List

This is the complete list of members for **DtLgrPbClock**, including all inherited members.

- **DtLgrPbClock**(DtClock *c)
- **DtLgrScalingClock**(DtClock *c)
- **getConstraint**()
- **getConstScaledTime**()
- **getIdealScaledTime**()
- **getRealTime**() [inline]
- **getTimeScale**()
- **idealOffset** [protected]
- **idealScaledToReal**(DtTime scaledTime)
- **myClock** [protected]
- **realToIdealScaled**(DtTime realTime)
- **setConstraint**(DtTime c)
- **setConstScaledTime**(DtTime t)
- **setIdealScaledTime**(DtTime t)
- **setTimeScale**(double ts) [virtual]
- **timeScale** [protected]
- **~DtLgrPbClock**() [virtual]
- **~DtLgrScalingClock**() [virtual]

DtLgrRecClock Member List

This is the complete list of members for **DtLgrRecClock**, including all inherited members.

- **DtLgrRecClock**(DtClock *c)
- **DtLgrScalingClock**(DtClock *c)
- **getIdealScaledTime**()
- **getRealTime**() [inline]
- **getTimeScale**()
- **idealOffset** [protected]
- **idealScaledToReal**(DtTime scaledTime)
- **myClock** [protected]
- **realToIdealScaled**(DtTime realTime)
- **setIdealScaledTime**(DtTime t)
- **setTimeScale**(double ts) [virtual]
- **timeScale** [protected]
- **~DtLgrRecClock**() [virtual]
- **~DtLgrScalingClock**() [virtual]

DtLgrScalingClock Member List

This is the complete list of members for **DtLgrScalingClock**, including all inherited members.

- **DtLgrScalingClock**(DtClock *c)
- **getIdealScaledTime**()
- **getRealTime**() [inline]
- **getTimeScale**()
- **idealOffset** [protected]
- **idealScaledToReal**(DtTime scaledTime)
- **myClock** [protected]
- **realToIdealScaled**(DtTime realTime)
- **setIdealScaledTime**(DtTime t)
- **setTimeScale**(double ts)=0 [protected, pure virtual]
- **timeScale** [protected]
- **~DtLgrScalingClock**() [virtual]

DtLgrState Member List

This is the complete list of members for **DtLgrState**, including all inherited members.

- **annotations()** [virtual]
- **clearFlags()** [virtual]
- **clock()** const [virtual]
- **dataScaling()** const [virtual]
- **DtLgrState**(DtExerciseConn *exConn, DtLgrPbClock *clock)
- **eofAction()** const [virtual]
- **exerciseConn()** const [virtual]
- **fetchFromClipboard**(DtString keyname) [virtual]
- **file()** const [virtual]
- **filePosition()** const [virtual]
- **heartBeat()** const [virtual]
- **mode()** const [virtual]
- **myAnnotations** [protected]
- **myClipboard** [protected]
- **myClock** [protected]
- **myDataScaling** [protected]
- **myEofAction** [protected]
- **myExConn** [protected]
- **myFile** [protected]
- **myFileLength** [protected]
- **myFilePosition** [protected]
- **myHeartbeat** [protected]
- **myMode** [protected]
- **myPacketLimit** [protected]
- **myPlayAction** [protected]
- **myPlaybackConstructMethod** [protected]
- **myRecordAction** [protected]
- **myRecordConstructMethod** [protected]
- **mySkipToEnd** [protected]
- **mySkipToStart** [protected]

- **myStateRep** [protected]
- **myTimeJumped** [protected]
- **myTimeout** [protected]
- **packetLimit**() const [virtual]
- **playAction**() const [virtual]
- **playbackFileCreator**() [virtual]
- **postToClipboard**(DtString keyname, void *data) [virtual]
- **recordAction**() const [virtual]
- **recordFileCreator**() [virtual]
- **removeFromClipboard**(DtString keyname) [virtual]
- **setDataScaling**(DtBoolean onOff) [virtual]
- **setEOFAction**(DtLgrMode e) [virtual]
- **setFile**(LtLoggerFile *f) [virtual]
- **setFilePosition**(DtLgrFilePosition pos) [virtual]
- **setHeartBeat**(double val) [virtual]
- **setMode**(DtLgrMode m) [virtual]
- **setPacketLimit**(int val) [virtual]
- **setPlayAction**(DtLgrMode m) [virtual]
- **setPlaybackFileCreator**(DtLoggerPlaybackConstructMethod m) [virtual]
- **setRecordAction**(DtLgrMode m) [virtual]
- **setRecordFileCreator**(DtLoggerRecordConstructMethod m) [virtual]
- **setSkipToEnd**() [virtual]
- **setSkipToStart**() [virtual]
- **setTimeJumped**(int state=2) [virtual]
- **setTimeout**(double val) [virtual]
- **skipToEnd**() const [virtual]
- **skipToStart**() const [virtual]
- **stateReps**() [virtual]
- **timeJumped**() const [virtual]
- **timeout**() const [virtual]
- **~DtLgrState**() [virtual]

DtLogger Member List

This is the complete list of members for **DtLogger**, including all inherited members.

- **addObject**(LtObjectType t, const char *keyname, DtString theParam) [virtual]
- **annotations**() const [virtual]
- **currentFile**() const [virtual]
- **dataScaling**() const [virtual]
- **DtLogger**(DtExerciseConn *exConn)
- **DtLogger**(int port, int exercise_id, int site_id, int application_num)
- **DtLogger**(int port, DtInetAddr destAddr, int exercise_id, int site_id, int application_num)
- **DtLogger**(DtSocket *skt, int exercise_id, int site_id, int application_num)
- **eofAction**() const [virtual]
- **exerciseConn**() const [virtual]
- **fetchFromClipboard**(DtString keyname) [virtual]
- **file**() const [virtual]
- **fileLength**() const [virtual]
- **filePosition**() const [virtual]
- **gotoEnd**() [virtual]
- **gotoStart**() [virtual]
- **heartBeat**() const [virtual]
- **idealTime**() const [virtual]
- **init**() [protected]
- **inPlayback**() const [virtual]
- **loadFileForPlayback**(char *filename) [virtual]
- **loadFileForRecording**(char *filename) [virtual]
- **mode**() const [virtual]
- **myClock** [protected]
- **myCreatedExConn** [protected]
- **myExConn** [protected]
- **myFilename** [protected]
- **myHeartbeat** [protected]
- **myMirror** [protected]
- **myPb** [protected]

- **myPbSend** [protected]
- **myPbSource** [protected]
- **myPipeline** [protected]
- **myRecLgrCtrlInfo** [protected]
- **myRecSend** [protected]
- **myRecSource** [protected]
- **myRecSrcParam** [protected]
- **myState** [protected]
- **myTimeout** [protected]
- **myTimeScale** [protected]
- **pause()** [virtual]
- **play()** [virtual]
- **playAction()** const [virtual]
- **postToClipboard**(DtString keyname, void *data) [virtual]
- **record()** [virtual]
- **recordAction()** const [virtual]
- **recordingLoggerControlInfo()** const [virtual]
- **registerObject**(const char *keyname, DtCreateMethod m) [virtual]
- **removeFromClipboard**(DtString keyname) [virtual]
- **removeObject**(LtObjectType t, const char *keyname) [virtual]
- **reset()** [virtual]
- **setDataScaling**(DtBoolean onOff) [virtual]
- **setDefaultRecordSourceParameter**(DtString param) [virtual]
- **setEofAction**(DtLgrMode e) [virtual]
- **setHeartBeat**(double val) [virtual]
- **setMirroring**(DtBoolean onOff) [virtual]
- **setMode**(DtLgrMode m) [protected]
- **setPacketLimit**(int val) [virtual]
- **setPlayAction**(DtLgrMode m) [virtual]
- **setPlaybackFileCreator**(DtLoggerPlaybackConstructMethod m) [virtual]
- **setRecordAction**(DtLgrMode m) [virtual]
- **setRecordFileCreator**(DtLoggerRecordConstructMethod m) [virtual]
- **setRecordingLoggerControlInfo**(DtBoolean onOff) [virtual]
- **setTime**(DtTime newTime) [virtual]
- **setTimeout**(double val) [virtual]
- **setTimeScale**(double timescale) [virtual]
- **setup**(char *) [protected]
- **skipTime**(DtTime amount) [virtual]

- [stop\(\)](#) [virtual]
- [tick\(\)](#) [virtual]
- [time\(\)](#) const [virtual]
- [timeout\(\)](#) const [virtual]
- [timeScale\(\)](#) const [virtual]
- [~DtLogger\(\)](#) [virtual]

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtLoggerStage Member List

This is the complete list of members for **DtLoggerStage**, including all inherited members.

- [activate\(\)](#)=0 [pure virtual]
- [deactivate\(\)](#)=0 [pure virtual]
- [DtLoggerStage](#)(const char *theName)
- [DtLoggerStage](#)(const DtLoggerStage &) [protected]
- [getName\(\)](#) const [virtual]
- [initSelf](#)(DtLgrState *theState)=0 [pure virtual]
- [isOk\(\)](#) const=0 [pure virtual]
- [myName](#) [protected]
- [printInfo\(\)](#) const=0 [pure virtual]
- [reset\(\)](#)=0 [pure virtual]
- [tick\(\)](#)=0 [pure virtual]
- [~DtLoggerStage\(\)](#) [virtual]

DtMotifGui Member List

This is the complete list of members for **DtMotifGui**, including all inherited members.

- **[aboutCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[aboutOkCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[aboutWidget](#)**
- **[addAnnotHelpCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[addAnnotOkCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[addAnnotTextWidget](#)**
- **[addAnnotTimeLabel](#)**
- **[addAnnotWidget](#)**
- **[addBrowseWidget](#)**
- **[addInputButton](#)**
- **[addInputCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[addInputOkCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[addInputSelectCallback](#)**(Widget w, void *num, XtPointer) [static]
- **[addInputWidget](#)**
- **[addOutputBrowseWidget](#)**
- **[addOutputButton](#)**
- **[addOutputCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[addOutputOkCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[addOutputSelectCallback](#)**(Widget w, void *num, XtPointer) [static]
- **[addOutputTextWidget](#)**
- **[addOutputWidget](#)**
- **[addParmTextWidget](#)**
- **[annotAddCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[annotCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[annotClearCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[annotCloseCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[annotCurFile](#)**
- **[annotGuiMapping](#)**
- **[annotHelpMenuCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **[annotImportCallback](#)**(Widget w, DtMotifGui *aif, XtPointer) [static]

- **annotList**
- **annotMenu**
- **annotModCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **annotOpenCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **annotRemoveCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **annotSaveasCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **annotSaveCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **annotText**
- **annotWidget**
- **arrangeForMode**(DtLgrMode m)
- **askYN**(char *prompt, int defChoice)
- **bmArray**
- **browserOpen**
- **buttonActive**
- **buttonPressed**
- **cachedMode**
- **cachedScale**
- **cachedTime**
- **colorWidget**(Widget w, int inverse)
- **configCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **configCancelCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **configHelpCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **configOkCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **configRadioCallback**(Widget w, int which, XmToggleButtonCallbackStruct *state) [static]
- **configWidget**
- **context**
- **createInterface**(Widget top)
- **currentFileField**
- **currentStatusField**
- **currentTimeField**
- **currentTimeLabel**
- **displayButtons**(DtLgrMode m)
- **DtLgrInterface**(DtLogger *logger) [inline]
- **DtMotifGui**(DtLogger *logger, DtBoolean advanced=DtFalse)
- **editPix**
- **eofActLoopRadio**
- **eofActPauseRadio**
- **eofActStopRadio**

- **EOFActToggleCallback**(Widget w, int n, XmToggleButtonCallbackStruct *cbs) [static]
- **EOFButton**
- **EOFCallback**(Widget, XtPointer, XtPointer) [static]
- **EOFLabel**
- **eofPix**
- **eofPixG**
- **fileBrowseCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **fileBrowseOkCallback**(Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs) [static]
- **fileBrowseWidget**
- **fileMenu**
- **getTsIndex**(float val)
- **heartbeatWidget**
- **helpCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **helpMenu**
- **importAnnotOkCallback**(Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs) [static]
- **importAnnotWidget**
- **initSelf**(DtLogger *theAIF)
- **inputList**
- **inputRemoveCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **inputSelNum**
- **jumpCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **jumpPix**
- **listSelectionCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **logoPix**
- **manInputCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **manInputMenu**
- **manInputWidget**
- **manOutputCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **manOutputMenu**
- **manOutputWidget**
- **menuBar**
- **modAnnotHelpCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **modAnnotOkCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **modAnnotTextWidget**
- **modAnnotTimeLabel**
- **modAnnotWidget**
- **myAdvanced**

- **myAnnotAddTime**
- **myAnnotCount**
- **myBrowser**
- **myLogger**
- **mySelectedAnnotation**
- **newCallback**(Widget, DtMotifGui *aif, XtPointer) [static]
- **newFileCancelCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **newFileOkCallback**(Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs) [static]
- **newFileWidget**
- **newMenu**
- **openActToggleBox**
- **openAnnotCancelCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **openAnnotOkCallback**(Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs) [static]
- **openAnnotWidget**
- **openCallback**(Widget, DtMotifGui *aif, XtPointer) [static]
- **openFileCancelCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **openFileOkCallback**(Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs) [static]
- **openFileWidget**
- **openMenu**
- **optionMenu**
- **outputList**
- **outputRemoveCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **outputSelNum**
- **pauseButton**
- **pauseCallback**(Widget, XtPointer, XtPointer) [static]
- **pauseLabel**
- **pausePix**
- **pausePixG**
- **pausePixPG**
- **pb**
- **pendingEOFAction**
- **pendingPlayAction**
- **pendingRecordAction**
- **playActPlayRadio**
- **playActStopRadio**
- **playActToggleCallback**(Widget w, int n, XmToggleButtonCallbackStruct *cbs) [static]

- **playButton**
- **playCallback**(Widget, XtPointer, XtPointer) [static]
- **playLabel**
- **playPix**
- **playPixG**
- **playPixP**
- **playPixPG**
- **playToggleBox**
- **processAnnotationFile**(const char *theFile)
- **quitCallback**(Widget, DtMotifGui *aif, XtPointer) [static]
- **recordActRecordRadio**
- **recordActStopRadio**
- **recordActToggleCallback**(Widget w, int n, XmToggleButtonCallbackStruct *cbs) [static]
- **recordButton**
- **recordCallback**(Widget, XtPointer, XtPointer) [static]
- **recordLabel**
- **recordPix**
- **recordPixG**
- **recordPixP**
- **recordPixPG**
- **removeInputButton**
- **removeOutputButton**
- **resetAnnotList**() [virtual]
- **saveasAnnotOkCallback**(Widget w, DtMotifGui *aif, XmFileSelectionBoxCallbackStruct *cbs) [static]
- **saveasAnnotWidget**
- **saveNoteList**(const char *filename)
- **setScaleCancelCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **setScaleHelpCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **setScaleOkCallback**(Widget w, DtMotifGui *aif, XmSelectionBoxCallbackStruct *c) [static]
- **setScaleWidget**
- **setTimeCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **setTimeCancelCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **setTimeHelpCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **setTimeMenu**
- **setTimeOkCallback**(Widget w, DtMotifGui *aif, XmSelectionBoxCallbackStruct *cbs) [static]
- **setTimeToQuit**(DtBoolean yesNo) [inline, virtual]

- **setTimeWidget**
- **shadeAndSensitizeButton**(Widget, Pixmap, Pixmap, Boolean)
- **shell**
- **showFilename**(char *f)
- **showMode**(DtLgrMode m)
- **showPositional**()
- **showScale**()
- **showTime**()
- **skipTimeCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **skipTimeCancelCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **skipTimeHelpCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **skipTimeMenu**
- **skipTimeOkCallback**(Widget w, DtMotifGui *aif, XmSelectionBoxCallbackStruct *cbs) [static]
- **skipTimeToggleCallback**(Widget w, int n, XmToggleButtonCallbackStruct *cbs) [static]
- **skipTimeWidget**
- **SOFButton**
- **SOFCallback**(Widget, XtPointer, XtPointer) [static]
- **SOFLabel**
- **sofPix**
- **sofPixG**
- **stopButton**
- **stopCallback**(Widget, XtPointer, XtPointer) [static]
- **stopLabel**
- **stopPix**
- **stopPixG**
- **tick**() [virtual]
- **timeoutWidget**
- **timeScaleCallback**(Widget w, DtMotifGui *aif, XtPointer) [static]
- **timeScaleMenu**
- **timeScaleSlider**
- **timeScaleValues**
- **timeScaleValuesCount**
- **timeToQuit**() [inline, virtual]
- **topForm**
- **topLevel**
- **topManager**
- **updateAnnotList**() [virtual]
- **warn**(char *warning)

- [~DtLgrInterface\(\)](#) [inline, virtual]
- [~DtMotifGui\(\)](#) [virtual]

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

DtNote Member List

This is the complete list of members for **DtNote**, including all inherited members.

- **DtNote**(DtTime theTime, DtString theMsg) [inline]
- **msg**() [inline, virtual]
- **setMsg**(char *m) [inline, virtual]
- **setTime**(DtTime t) [inline, virtual]
- **time**() [inline, virtual]

DtSendObj Member List

This is the complete list of members for **DtSendObj**, including all inherited members.

- **activate()**=0 [pure virtual]
- **deactivate()**=0 [pure virtual]
- **DtLoggerStage**(const char *theName)
- **DtLoggerStage**(const DtLoggerStage &) [protected]
- **DtSendObj**(const char *theName)
- **getName()** const [virtual]
- **initSelf**(DtLgrState *theState)=0 [pure virtual]
- **isOk()** const [virtual]
- **myName** [protected]
- **myStatus** [protected]
- **printInfo()** const=0 [pure virtual]
- **processData**(const DtList *packetList) [virtual]
- **reset()** [inline, virtual]
- **sendPacket**(const LtPacketInfo *packet)=0 [pure virtual]
- **tick()** [inline, virtual]
- **~DtLoggerStage()** [virtual]
- **~DtSendObj()** [virtual]

DtSendToFile Member List

This is the complete list of members for **DtSendToFile**, including all inherited members.

- **activate()** [virtual]
- **create**(DtString) [static]
- **deactivate()** [virtual]
- **DtLoggerStage**(const char *theName)
- **DtLoggerStage**(const DtLoggerStage &) [protected]
- **DtSendObj**(const char *theName)
- **DtSendToFile**(DtString theName)
- **getName()** const [virtual]
- **initSelf**(DtLgrState *theState) [virtual]
- **isOk()** const [virtual]
- **myFileName** [protected]
- **myFp** [protected]
- **myName** [protected]
- **myRecFile** [protected]
- **myState** [protected]
- **myStatus** [protected]
- **printInfo()** const [virtual]
- **processData**(const DtList *packetList) [virtual]
- **reset()** [inline, virtual]
- **sendPacket**(const LtPacketInfo *packet) [inline, virtual]
- **tick()** [inline, virtual]
- **~DtLoggerStage()** [virtual]
- **~DtSendObj()** [virtual]
- **~DtSendToFile()**

DtSourceObj Member List

This is the complete list of members for **DtSourceObj**, including all inherited members.

- **activate**(DtTime startTime)=0 [pure virtual]
- **activate**() [inline, virtual]
- **deactivate**()=0 [pure virtual]
- **DtLoggerStage**(const char *theName)
- **DtLoggerStage**(const DtLoggerStage &) [protected]
- **DtSourceObj**(const char *theName)
- **generatePacket**(DtTime currentTime, DtTime *nextTime)=0 [pure virtual]
- **generatePackets**(DtTime currentTime, DtList *packetList) [virtual]
- **getName**() const [virtual]
- **initSelf**(DtLgrState *theState)=0 [pure virtual]
- **isOk**() const [virtual]
- **myName** [protected]
- **myStatus** [protected]
- **printInfo**() const [virtual]
- **reset**()=0 [pure virtual]
- **tick**() [inline, virtual]
- **~DtLoggerStage**() [virtual]
- **~DtSourceObj**() [virtual]

LtLoggerFile Member List

This is the complete list of members for **LtLoggerFile**, including all inherited members.

- **closeFile()** [protected, virtual]
- **filePtr()** const [virtual]
- **isOk()** const [virtual]
- **LtLoggerFile**(const char *fileName, const char *flags) [protected]
- **myFileName** [protected]
- **myFp** [protected]
- **myStatus** [protected]
- **name()** const [virtual]
- **openFile**(const char *fileName, const char *flags) [protected, virtual]
- **~LtLoggerFile()** [virtual]

LtLoggerPlaybackFile Member List

This is the complete list of members for **LtLoggerPlaybackFile**, including all inherited members.

- **atEnd()** [virtual]
- **atStart()** [virtual]
- **checkMagicNumber**(const char *filename) [static]
- **checkMagicNumber()** [protected, virtual]
- **closeFile()** [protected, virtual]
- **filePtr()** const [virtual]
- **fileTimeBackwards**(DtTime *t) [virtual]
- **fileTimeForwards**(DtTime *t) [virtual]
- **firstPacketTime**(DtTime *t) [virtual]
- **isOk()** const [virtual]
- **jumpToEnd()** [virtual]
- **jumpToStart()** [virtual]
- **lastPacketTime**(DtTime *t) [virtual]
- **LtLoggerFile**(const char *fileName, const char *flags) [protected]
- **LtLoggerPlaybackFile**(const char *FileName)
- **myEndPos** [protected]
- **myFileName** [protected]
- **myFirstTime** [protected]
- **myFp** [protected]
- **myLastTime** [protected]
- **myStartPos** [protected]
- **myStatus** [protected]
- **name()** const [virtual]
- **openFile**(const char *fileName, const char *flags) [protected, virtual]
- **readPacket()** [virtual]
- **readPacketBackwards()** [virtual]
- **readPacketData**(void *buffer, int bytesToRead) [protected, virtual]
- **readPacketHeader**(LtNetPacketHeader *header) [protected, virtual]
- **readPacketTail**(LtNetPacketTail *tail) [protected, virtual]
- **seekBackward**(DtTime t) [virtual]

- **seekForward**(DtTime t) [virtual]
- **seekTime**(DtTime t) [virtual]
- **skipPacketBackwards**() [virtual]
- **skipPacketForwards**() [virtual]
- **~LtLoggerFile**() [virtual]
- **~LtLoggerPlaybackFile**() [virtual]

[Main Page](#) [Class Hierarchy](#) [Alphabetical List](#) [Compound List](#) [File List](#) [Compound Members](#)
[File Members](#)

LtLoggerRecordFile Member List

This is the complete list of members for **LtLoggerRecordFile**, including all inherited members.

- **closeFile()** [protected, virtual]
- **filePtr()** const [virtual]
- **isOk()** const [virtual]
- **LtLoggerFile**(const char *fileName, const char *flags) [protected]
- **LtLoggerRecordFile**(const char *FileName)
- **myFileName** [protected]
- **myFp** [protected]
- **myStatus** [protected]
- **name()** const [virtual]
- **openFile**(const char *fileName, const char *flags) [protected, virtual]
- **writeMagicNumber()** [protected, virtual]
- **writePacket**(LtPacketInfo *packetInfo) [virtual]
- **writePacket**(void *buffer, int length, double time) [virtual]
- **~LtLoggerFile()** [virtual]
- **~LtLoggerRecordFile()** [virtual]

LtNetInteractionRecord Member List

This is the complete list of members for **LtNetInteractionRecord**, including all inherited members.

- **[classHandle](#)**
- **[numParams](#)**
- **[numValueBytes](#)**

LtNetPacketHeader Member List

This is the complete list of members for [LtNetPacketHeader](#), including all inherited members.

- [arrivalTime](#)
- [numBytes](#)
- [padding](#)

LtNetPacketTail Member List

This is the complete list of members for [LtNetPacketTail](#), including all inherited members.

- [numBytes](#)

LtNetRemovalRecord Member List

This is the complete list of members for [LtNetRemovalRecord](#), including all inherited members.

- [classHandle](#)
- [nameSize](#)
- [objectId](#)
- [pad1](#)
- [pad2](#)

LtNetUpdateRecord Member List

This is the complete list of members for [LtNetUpdateRecord](#), including all inherited members.

- [classHandle](#)
- [nameSize](#)
- [numAttrs](#)
- [numValueBytes](#)
- [objectId](#)
- [padding](#)

LtPacketInfo Member List

This is the complete list of members for **LtPacketInfo**, including all inherited members.

- **arrivalTime**() const [virtual]
- **LtPacketInfo**()
- **LtPacketInfo**(const LtPacketInfo &orig)
- **myArrivalTime** [protected]
- **myPacket** [protected]
- **myPacketKind** [protected]
- **mySize** [protected]
- **operator**=(const LtPacketInfo &orig)
- **packet**() const [virtual]
- **packetKind**() const [virtual]
- **readPacket**(const char *buffer) [virtual]
- **setArrivalTime**(DtTime theTime) [virtual]
- **setPacket**(char *thePacket) [virtual]
- **setPacketKind**(LtPacketKind theKind) [virtual]
- **setSize**(int theSize) [virtual]
- **size**() const [virtual]
- **writePacket**(char *buffer) [virtual]
- **~LtPacketInfo**() [virtual]

Logger 3.7 DIS Classes Graphical Class Hierarchy

[Go to the textual class hierarchy](#)

