HASAS Reference Manual

1.7.0

Generated by Doxygen 1.3.3

Sun Oct 26 19:11:18 2003

Contents

Chapter 1

HASAS Main Page

HASAS - HydroAcoustic Signal Analysis System

Author:
: Jussi Laako

  Date
  : 2003/07/12 20:21:13

Overview
: HASAS is signal analysis software for passive sonar systems.

Licensing

Chapter 2

HASAS Namespace Index

2.1 HASAS Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

HASAS Hierarchical Index

3.1 HASAS Class Hierarchy

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

Chapter 4

HASAS Compound Index

4.1 HASAS Compound List

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

Chapter 5

HASAS File Index

5.1 HASAS File List

Here is a list of all files with brief descriptions:

Chapter 6

HASAS Namespace Documentation

6.1 Magick Namespace Reference

6.2 std Namespace Reference

Chapter 7

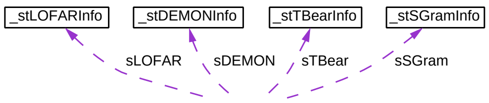
HASAS Class Documentation

7.1 _uPicInfo Union Reference

Combination of information in all .inf files.

`#include <ConvPic.hh>`

Collaboration diagram for _uPicInfo:



Public Attributes

- **stLOFARInfo sLOFAR**
- **stDEMONInfo sDEMON**
- **stSGramInfo sSGram**
- **stTBearInfo sTBear**

7.1.1 Detailed Description

Combination of information in all .inf files.

Definition at line 109 of file ConvPic.hh.

7.1.2 Member Data Documentation

7.1.2.1 stLOFARInfo _uPicInfo::sLOFAR

Definition at line 111 of file ConvPic.hh.

Referenced by clConvPic::DrawLOFARInfo(), and clConvPic::LoadLOFARInfo().

7.1.2.2 stDEMONInfo _uPicInfo::sDEMON

Definition at line 112 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), and clConvPic::LoadDEMONInfo().

7.1.2.3 stSGramInfo _uPicInfo::sSGram

Definition at line 113 of file ConvPic.hh.

Referenced by clConvPic::DrawSGramInfo(), and clConvPic::LoadSGramInfo().

7.1.2.4 stTBearInfo _uPicInfo::sTBear

Definition at line 114 of file ConvPic.hh.

Referenced by clConvPic::DrawTBearInfo(), and clConvPic::LoadTBearInfo().

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

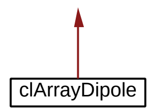
- **ConvPic.hh**

7.2 clArrayBase Class Reference

Base class for array operations.

#include <ArrayBase.hh>

Inheritance diagram for clArrayBase:



Public Member Functions

- **clArrayBase** ()
- ~**clArrayBase** ()
- void **EnableDebug** ()
- void **DisableDebug** ()
- void **SetSampleRate** (int)

    Set sample rate.

- void **SetSoundSpeed** (GDT)

    Set sound speed.

- void **SetShading** (GDT ∗, int, long)

    Generate shading coefficients.

Public Attributes

- bool **bDebug**
- GDT **fSoundSpeed**
- GDT **fSecsPerMeter**

    1.0 / soundspeed

- GDT **fSampleSpacing**

    1.0 / samplerate

- clDSPOp **DSP**

7.2.1 Detailed Description

Base class for array operations.

Definition at line 46 of file ArrayBase.hh.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 clArrayBase::clArrayBase () `[inline]`

Definition at line 54 of file ArrayBase.hh.

References bDebug.

7.2.2.2 clArrayBase::~clArrayBase () `[inline]`

Definition at line 55 of file ArrayBase.hh.

7.2.3 Member Function Documentation

7.2.3.1 void clArrayBase::EnableDebug () `[inline]`

Definition at line 56 of file ArrayBase.hh.

References bDebug.

Referenced by clArrayDipole::Initialize().

7.2.3.2 void clArrayBase::DisableDebug () `[inline]`

Definition at line 57 of file ArrayBase.hh.

References bDebug.

7.2.3.3 void clArrayBase::SetSampleRate (int)

Set sample rate.

$$\Delta t_s = \frac{1}{f_s}$$

Parameters:
iSampleRate Samplerate

Definition at line 34 of file ArrayBase.cc.

References bDebug, fSampleSpacing, and GDT.

Referenced by clArrayDipole::Initialize().

7.2.3.4 void clArrayBase::SetSoundSpeed (GDT)

Set sound speed.

$$c^{-1} = \frac{1}{c}$$

Parameters:

fSndSpeed Speed of sound in m/s

Definition at line 41 of file ArrayBase.cc.

References bDebug, fSecsPerMeter, fSoundSpeed, and GDT.

Referenced by clArrayDipole::Initialize().

7.2.3.5 void clArrayBase::SetShading (GDT *, int, long)

Generate shading coefficients.

Parameters:

fpCoeffs Shading coefficients

iShadeType Type of shade window

lSensCount Number of sensors

Definition at line 49 of file ArrayBase.cc.

References AB_KBWIN_ALPHA, AB_SHADE_BLACKMAN, AB_SHADE_KAISER_BESSEL, AB_SHADE_RECTANGLE, bDebug, cpaShadingTypes, DSP, and GDT.

7.2.4 Member Data Documentation

7.2.4.1 bool clArrayBase::bDebug

Definition at line 49 of file ArrayBase.hh.

Referenced by clArrayBase(), DisableDebug(), EnableDebug(), clArrayDipole::Initialize(), clArrayDipole::SetLowFreqLimit(), SetSampleRate(), SetShading(), and SetSoundSpeed().

7.2.4.2 GDT clArrayBase::fSoundSpeed

Definition at line 50 of file ArrayBase.hh.

Referenced by SetSoundSpeed().

7.2.4.3 GDT clArrayBase::fSecsPerMeter

1.0 / soundspeed

Definition at line 51 of file ArrayBase.hh.

Referenced by clArrayDipole::Initialize(), and SetSoundSpeed().

7.2.4.4 GDT clArrayBase::fSampleSpacing

1.0 / samplerate

Definition at line 52 of file ArrayBase.hh.

Referenced by clArrayDipole::GetDelaySamples(), clArrayDipole::GetMaxDelay(), and SetSampleRate().

7.2.4.5 clDSPOp clArrayBase::DSP

Definition at line 53 of file ArrayBase.hh.

Referenced by clArrayDipole::AddData(), clArrayDipole::GetAudioData(), clArrayDipole::GetFilteredData(), clArrayDipole::GetRawData(), clArrayDipole::Initialize(), and SetShading().

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

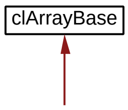
- **ArrayBase.hh**
- **ArrayBase.cc**

7.3 clArrayDipole Class Reference

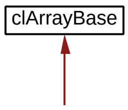
Class for dipole array operations.

`#include <ArrayDipole.hh>`

Inheritance diagram for clArrayDipole:



Collaboration diagram for clArrayDipole:



Public Member Functions

- **clArrayDipole** ()
- ~**clArrayDipole** ()
- bool **Initialize** (GDT, GDT, int, long, GDT, bool)

  Initialize array.

- void **Uninitialize** ()

  Uninitialize array.

- GDT * **GetRawPtr** (long)

  Get pointer to data buffer.

- GDT * **GetFiltPtr** (long)
- GDT * **GetHFPtr** (long)
- long **GetMaxDelay** ()

  Get maximum delay in samples.

- GDT **GetArrayFreq** ()

  Get upper frequency limit of array.

- GDT **GetDelay** (GDT)

  Get delay for specified direction (rad).

- GDT **GetDelayTime** (long, GDT)

  Get delay length in seconds for specified sensor to specified direction (rad).

- long **GetDelaySamples** (long, GDT)

  Get delay length in samples for specified sensor to specified direction (rad).

- void **SetLowFreqLimit** (GDT)

  Set lower frequency limit.

- void **AddData** (const GDT *, long, long)

  Add data to processing buffer.

- void **GetRawData** (GDT **, GDT)

  Get raw (non filtered) data for specified direction.

- GDT * **GetRawDataPtr** (long, GDT)
- void **GetFilteredData** (GDT **, GDT)

  Get filtered data for specified direction.

- GDT * **GetFilteredDataPtr** (long, GDT)
- void **GetAudioData** (GDT *, GDT, bool)

  Get audio data for specified direction.

Private Attributes

- bool **bInitialized**
- long **lWindowSize**

  for filtering and array processing = out size

- long **lBufferSize**

  *total size of history buffer n * WindowSize*

- long **lBaseIdx**

  Start of non-delayed window.

- int **iSampleRate**
- GDT **fSpacing**
- GDT **fDelay**
- clDSPAlloc **RawBuf** [2]

  Raw data.

- clDSPAlloc **FiltBuf** [2]

  Filtered data (BP filtered).

- clDSPAlloc **HFBuf** [2]
  High frequency data (HP filtered).
- clFilter **FilterBank** [4]
  0,1 BP filters, 2,3 HP filters

7.3.1 Detailed Description

Class for dipole array operations.

Definition at line 37 of file ArrayDipole.hh.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 clArrayDipole::clArrayDipole ()

Definition at line 30 of file ArrayDipole.cc.

References bInitialized.

7.3.2.2 clArrayDipole::~clArrayDipole ()

Definition at line 36 of file ArrayDipole.cc.

References bInitialized, and Uninitialize().

7.3.3 Member Function Documentation

7.3.3.1 bool clArrayDipole::Initialize (GDT, GDT, int, long, GDT, bool)

Initialize array.

$$\Delta T_d = dc^{-1}$$

Parameters:

fSensorSpacing Sensor spacing (m)

fSndSpeed Speed of sound (m/s)

iSRate Samplerate

lWinSize Size of window

fLowFreqLimit Low frequency limit (Hz)

bEnableDebug Enable debug printout

Returns:

Success

Definition at line 42 of file ArrayDipole.cc.

References clArrayBase::bDebug, bInitialized, clArrayBase::DSP, clArrayBase::EnableDebug(), fDelay, FiltBuf, FilterBank, clArrayBase::fSecsPerMeter, fSpacing, GDT, GetArrayFreq(), GetMaxDelay(), HFBuf, iSampleRate, lBaseIdx, lBufferSize, lWindowSize, RawBuf, clArrayBase::SetSampleRate(), and clArrayBase::SetSoundSpeed().

Referenced by clDirection::InitArray(), and main().

7.3.3.2 void clArrayDipole::Uninitialize ()

Uninitialize array.

Definition at line 96 of file ArrayDipole.cc.

References bInitialized, and FilterBank.

Referenced by main(), and ~clArrayDipole().

7.3.3.3 GDT * clArrayDipole::GetRawPtr (long)

Get pointer to data buffer.

Parameters:
 lSensor Number of sensor

Returns:
 Pointer to data buffer

Definition at line 109 of file ArrayDipole.cc.

References GDT, lBaseIdx, and RawBuf.

Referenced by clCorrDipole::AddData().

7.3.3.4 GDT * clArrayDipole::GetFiltPtr (long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 116 of file ArrayDipole.cc.

References FiltBuf, GDT, and lBaseIdx.

Referenced by clCorrDipole::AddData(), and clBeamDipole::AddData().

7.3.3.5 GDT * clArrayDipole::GetHFPtr (long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 123 of file ArrayDipole.cc.

References GDT, HFBuf, and lBaseIdx.

7.3.3.6 long clArrayDipole::GetMaxDelay ()

Get maximum delay in samples.

$$\Delta T_{max} = \frac{\Delta T_d}{\Delta t_s}$$

Returns:
Maximum delay in samples

Definition at line 130 of file ArrayDipole.cc.

References fDelay, clArrayBase::fSampleSpacing, and GDT.

Referenced by clBeamDipole::clBeamDipole(), clCorrDipole::clCorrDipole(), and Initialize().

7.3.3.7 GDT clArrayDipole::GetArrayFreq ()

Get upper frequency limit of array.

$$f_a = \frac{\Delta T_d^{-1}}{2}$$

or

$$f_a = \frac{c^{-1}d}{2}$$

Returns:
Upper frequency limit of array (Hz)

Definition at line 136 of file ArrayDipole.cc.

References fDelay, and GDT.

Referenced by Initialize(), and SetLowFreqLimit().

7.3.3.8 GDT clArrayDipole::GetDelay (GDT)

Get delay for specified direction (rad).

$$\Delta T = \sin(\theta)\Delta T_d$$

or

$$\Delta T = \frac{\sin(\theta)d}{c}$$

Parameters:
fDirection Direction (rad)

Returns:
Delay (time)

Definition at line 142 of file ArrayDipole.cc.

References fDelay, and GDT.

Referenced by GetDelaySamples(), and GetDelayTime().

7.3.3.9 GDT clArrayDipole::GetDelayTime (long, GDT)

Get delay length in seconds for specified sensor to specified direction (rad).

$$\Delta T_n = \Delta T n$$

Parameters:
lSensor Sensor number

fDirection Direction (rad)

Returns:
Delay (time)

Definition at line 148 of file ArrayDipole.cc.

References GDT, and GetDelay().

Referenced by main().

7.3.3.10 long clArrayDipole::GetDelaySamples (long, GDT)

Get delay length in samples for specified sensor to specified direction (rad).

Parameters:
lSensor Sensor number

fDirection Direction (rad)

Returns:
Delay (samples)

Definition at line 163 of file ArrayDipole.cc.

References clArrayBase::fSampleSpacing, GDT, and GetDelay().

Referenced by GetAudioData(), GetFilteredData(), GetFilteredDataPtr(), GetRawData(), GetRawDataPtr(), main(), clCorrDipole::Process(), and clBeamDipole::Process().

7.3.3.11 void clArrayDipole::SetLowFreqLimit (GDT)

Set lower frequency limit.

Parameters:

fLowFreqLimit Lower frequency limit (Hz)

Definition at line 180 of file ArrayDipole.cc.

References clArrayBase::bDebug, FilterBank, GDT, GetArrayFreq(), and i-SampleRate.

7.3.3.12 void clArrayDipole::AddData (const GDT *, long, long)

Add data to processing buffer.

Note:

There must be enough data to fill each channel with lWindowSize amount of data.

Parameters:

fpSource Source vector

lStartIndex Starting channel index

lChannels Number of channels

Definition at line 198 of file ArrayDipole.cc.

References clArrayBase::DSP, FiltBuf, FilterBank, GDT, HFBuf, lBufferSize, lWindowSize, and RawBuf.

Referenced by clDirection::ReceiveData().

7.3.3.13 void clArrayDipole::GetRawData (GDT **, GDT)

Get raw (non filtered) data for specified direction.

Parameters:

fpaDest Destination vectors (array of pointers)

fDirection Direction (rad)

Definition at line 241 of file ArrayDipole.cc.

References clArrayBase::DSP, GDT, GetDelaySamples(), lBaseIdx, lWindow-Size, and RawBuf.

7.3.3.14 GDT * clArrayDipole::GetRawDataPtr (long, GDT)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters:

lSensor Sensor number

fDirection Direction (rad)

Returns:

Pointer to data

Definition at line 257 of file ArrayDipole.cc.

References GDT, GetDelaySamples(), lBaseIdx, and RawBuf.

7.3.3.15 void clArrayDipole::GetFilteredData (GDT **, GDT)

Get filtered data for specified direction.

Parameters:

fpaDest Destination vectors (array of pointers)

fDirection Direction (rad)

Definition at line 268 of file ArrayDipole.cc.

References clArrayBase::DSP, FiltBuf, GDT, GetDelaySamples(), lBaseIdx, and lWindowSize.

7.3.3.16 GDT * clArrayDipole::GetFilteredDataPtr (long, GDT)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters:

lSensor Sensor number

fDirection Direction (rad)

Returns:

Pointer to data

Definition at line 284 of file ArrayDipole.cc.

References FiltBuf, GDT, GetDelaySamples(), and lBaseIdx.

7.3.3.17 void clArrayDipole::GetAudioData (GDT *, GDT, bool)

Get audio data for specified direction.

Parameters:

fpDest Destination vector

fDirection Direction (rad)

bFullBand Full bandwidth or array-limited

Definition at line 295 of file ArrayDipole.cc.

References clArrayBase::DSP, FiltBuf, GDT, GetDelaySamples(), HFBuf, l-BaseIdx, and lWindowSize.

7.3.4 Member Data Documentation

7.3.4.1 bool clArrayDipole::bInitialized [private]

Definition at line 39 of file ArrayDipole.hh.

Referenced by clArrayDipole(), Initialize(), Uninitialize(), and ~clArray-Dipole().

7.3.4.2 long clArrayDipole::lWindowSize [private]

for filtering and array processing = out size

Definition at line 40 of file ArrayDipole.hh.

Referenced by AddData(), GetAudioData(), GetFilteredData(), GetRaw-Data(), and Initialize().

7.3.4.3 long clArrayDipole::lBufferSize [private]

total size of history buffer n * WindowSize

Definition at line 41 of file ArrayDipole.hh.

Referenced by AddData(), and Initialize().

7.3.4.4 long clArrayDipole::lBaseIdx [private]

Start of non-delayed window.

Definition at line 42 of file ArrayDipole.hh.

Referenced by GetAudioData(), GetFilteredData(), GetFilteredDataPtr(), Get-FiltPtr(), GetHFPtr(), GetRawData(), GetRawDataPtr(), GetRawPtr(), and Initialize().

7.3.4.5 int clArrayDipole::iSampleRate [private]

Definition at line 43 of file ArrayDipole.hh.

Referenced by Initialize(), and SetLowFreqLimit().

7.3.4.6 GDT clArrayDipole::fSpacing [private]

Definition at line 44 of file ArrayDipole.hh.

Referenced by Initialize().

7.3.4.7 GDT clArrayDipole::fDelay [private]

Definition at line 45 of file ArrayDipole.hh.

Referenced by GetArrayFreq(), GetDelay(), GetMaxDelay(), and Initialize().

7.3.4.8 clDSPAlloc clArrayDipole::RawBuf[2] [private]

Raw data.

Definition at line 47 of file ArrayDipole.hh.

Referenced by AddData(), GetRawData(), GetRawDataPtr(), GetRawPtr(), and Initialize().

7.3.4.9 clDSPAlloc clArrayDipole::FiltBuf[2] [private]

Filtered data (BP filtered).

Definition at line 49 of file ArrayDipole.hh.

Referenced by AddData(), GetAudioData(), GetFilteredData(), GetFilteredDataPtr(), GetFiltPtr(), and Initialize().

7.3.4.10 clDSPAlloc clArrayDipole::HFBuf[2] [private]

High frequency data (HP filtered).

Definition at line 51 of file ArrayDipole.hh.

Referenced by AddData(), GetAudioData(), GetHFPtr(), and Initialize().

7.3.4.11 clFilter clArrayDipole::FilterBank[4] [private]

0,1 BP filters, 2,3 HP filters

Definition at line 53 of file ArrayDipole.hh.

Referenced by AddData(), Initialize(), SetLowFreqLimit(), and Uninitialize().

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

- **ArrayDipole.hh**
- **ArrayDipole.cc**

7.4 clArraySensor Class Reference

Class representing sensor in sensor array for frequency domain beamforming.

`#include <ArraySensor.hh>`

Public Member Functions

- **clArraySensor** ()
- ~**clArraySensor** ()
- void **EnableDebug** ()
- void **DisableDebug** ()
- bool **Initialize** (GDT, GDT, long)

  Initialize sensor.

- void **SetSampleRate** (GDT)

  Set sample rate.

- void **SetSoundSpeed** (GDT)

  Set sound speed.

- void **SetArrayFrequency** (GDT)

  Set design frequency of array.

- void **SetShading** (GDT)

  Set shading coefficient.

- void **SetDirection** (GDT, bool=true)

  Set direction (in rad).

- void **Put** (const GDT *, long)

  Put sensor data into sensor processor.

- bool **Get** (GDT *, long)

  Get delayed data from the sensor processor.

Protected Member Functions

- void **SinCosC** (float, float *, float *)
- void **SinCosC** (double, double *, double *)

Protected Attributes

- bool **bDebug**
- long **lWindowSize**

  Processing window size.

- long **lProcCount**

  Number of samples per processing.

- GDT **fPI**
- GDT **fArrayFrequency**

  Array frequency.

- GDT **fSoundSpeed**

  Speed of sound.

- GDT **fSecsPerMeter**

  1.0 / soundspeed

- GDT **fSampleRate**

  Samplerate.

- GDT **fSampleSpacing**

  1.0 / samplerate

- GDT **fFreqResolution**

  Frequency resolution.

- GDT **fShadeCoeff**

  Shading coefficient.

- GDT **fLeftDistance**

  Left side sensor distance.

- GDT **fRightDistance**

  Right side sensor distance.

- GDT **fLeftTime**

  Left side sensor delay.

- GDT **fRightTime**

  Right side sensor delay.

- GDT **fDirection**

  Beam direction for delay.

- clDSPOp **DSP**
- clFilter **Filter**
- clDSPVector< GDT > **dspvInData**
- clDSPVector< GDT > **dspvProcData**
- clDSPVector< GDT > **dspvOutData**

7.4.1 Detailed Description

Class representing sensor in sensor array for frequency domain beamforming.

Definition at line 38 of file ArraySensor.hh.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 clArraySensor::clArraySensor ()

Definition at line 56 of file ArraySensor.cc.

References bDebug, fPI, and fShadeCoeff.

7.4.2.2 clArraySensor::~clArraySensor () `[inline]`

Definition at line 66 of file ArraySensor.hh.

7.4.3 Member Function Documentation

7.4.3.1 void clArraySensor::SinCosC (float, float *, float *) `[inline, protected]`

Definition at line 30 of file ArraySensor.cc.

Referenced by SetDirection().

7.4.3.2 void clArraySensor::SinCosC (double, double *, double *) `[inline, protected]`

Definition at line 43 of file ArraySensor.cc.

7.4.3.3 void clArraySensor::EnableDebug () `[inline]`

Definition at line 67 of file ArraySensor.hh.

References bDebug.

Referenced by clFreqBeamDipole::SetDebug().

7.4.3.4 void clArraySensor::DisableDebug () [inline]

Definition at line 68 of file ArraySensor.hh.

References bDebug.

Referenced by clFreqBeamDipole::SetDebug().

7.4.3.5 bool clArraySensor::Initialize (GDT, GDT, long)

Initialize sensor.

Frequency resolution is $\frac{f_{Nyquist}}{N+1}$.

Parameters:

fLDistance Left side sensor distance to end of the array (m)

fRDistance Right side sensor distance to end of the array (m)

lWinSize Processing window size

Returns:

Success

Definition at line 64 of file ArraySensor.cc.

References bDebug, dspvProcData, Filter, fLeftDistance, fRightDistance, GDT, lProcCount, and lWindowSize.

Referenced by clFreqBeamLine::Initialize(), and clFreqBeamDipole::Initialize().

7.4.3.6 void clArraySensor::SetSampleRate (GDT)

Set sample rate.

$$\Delta t_s = \frac{1}{f_s}$$

Parameters:

fFs Samplerate

Definition at line 87 of file ArraySensor.cc.

References bDebug, fFreqResolution, fSampleRate, fSampleSpacing, GDT, and lWindowSize.

Referenced by clFreqBeamLine::Initialize(), and clFreqBeamDipole::Initialize().

7.4.3.7 void clArraySensor::SetSoundSpeed (GDT)

Set sound speed.

$$c^{-1} = \frac{1}{c}$$

Parameters:
fSndSpeed Speed of sound in m/s

Definition at line 101 of file ArraySensor.cc.

References bDebug, fLeftDistance, fLeftTime, fRightDistance, fRightTime, f-SecsPerMeter, fSoundSpeed, and GDT.

Referenced by clFreqBeamDipole::SetSoundSpeed().

7.4.3.8 void clArraySensor::SetArrayFrequency (GDT)

Set design frequency of array.

$\lambda/2$

Parameters:
fFa Array frequency (Hz)

Definition at line 116 of file ArraySensor.cc.

References bDebug, and fArrayFrequency.

Referenced by clFreqBeamDipole::SetSoundSpeed().

7.4.3.9 void clArraySensor::SetShading (GDT)

Set shading coefficient.

Parameters:
fCoeff Shading coefficient

Definition at line 123 of file ArraySensor.cc.

References bDebug, and fShadeCoeff.

Referenced by clFreqBeamLine::Initialize().

7.4.3.10 void clArraySensor::SetDirection (GDT, bool = true)

Set direction (in rad).

$$\varphi_f = \sin(\theta)\pi\left(\frac{\frac{1}{c}d}{\frac{1}{2f}}\right)$$

$$C_f = \cos(\varphi_f) + j\sin(\varphi_f)$$

Parameters:

fDir Direction

bLowPass Low pass filter at array frequency

Definition at line 130 of file ArraySensor.cc.

References bDebug, fArrayFrequency, fFreqResolution, Filter, fLeftTime, fPI, fRightTime, GDT, lWindowSize, and SinCosC().

Referenced by clFreqBeamDipole::SetDirection().

7.4.3.11 void clArraySensor::Put (const GDT *, long)

Put sensor data into sensor processor.

Parameters:

fpSrc Source vector

lSrcCount Number of samples in source vector

Definition at line 172 of file ArraySensor.cc.

References dspvInData, dspvOutData, dspvProcData, Filter, and fShadeCoeff.

Referenced by clFreqBeamDipole::Put().

7.4.3.12 bool clArraySensor::Get (GDT *, long)

Get delayed data from the sensor processor.

Parameters:

fpDest Destination vector

lDestCount Number of samples to put into destination vector

Returns:

Success

Definition at line 187 of file ArraySensor.cc.

References dspvOutData.

Referenced by clFreqBeamDipole::Get().

7.4.4 Member Data Documentation

7.4.4.1 bool clArraySensor::bDebug `[protected]`

Definition at line 41 of file ArraySensor.hh.

Referenced by clArraySensor(), DisableDebug(), EnableDebug(), Initialize(), SetArrayFrequency(), SetDirection(), SetSampleRate(), SetShading(), and SetSoundSpeed().

7.4.4.2 long clArraySensor::lWindowSize `[protected]`

Processing window size.

Definition at line 42 of file ArraySensor.hh.

Referenced by Initialize(), SetDirection(), and SetSampleRate().

7.4.4.3 long clArraySensor::lProcCount `[protected]`

Number of samples per processing.

Definition at line 43 of file ArraySensor.hh.

Referenced by Initialize().

7.4.4.4 GDT clArraySensor::fPI `[protected]`

Definition at line 44 of file ArraySensor.hh.

Referenced by clArraySensor(), and SetDirection().

7.4.4.5 GDT clArraySensor::fArrayFrequency `[protected]`

Array frequency.

Definition at line 45 of file ArraySensor.hh.

Referenced by SetArrayFrequency(), and SetDirection().

7.4.4.6 GDT clArraySensor::fSoundSpeed `[protected]`

Speed of sound.

Definition at line 46 of file ArraySensor.hh.

Referenced by SetSoundSpeed().

7.4.4.7 GDT clArraySensor::fSecsPerMeter `[protected]`

1.0 / soundspeed

Definition at line 47 of file ArraySensor.hh.

Referenced by SetSoundSpeed().

7.4.4.8 GDT clArraySensor::fSampleRate [protected]

Samplerate.

Definition at line 48 of file ArraySensor.hh.

Referenced by SetSampleRate().

7.4.4.9 GDT clArraySensor::fSampleSpacing [protected]

1.0 / samplerate

Definition at line 49 of file ArraySensor.hh.

Referenced by SetSampleRate().

7.4.4.10 GDT clArraySensor::fFreqResolution [protected]

Frequency resolution.

Definition at line 50 of file ArraySensor.hh.

Referenced by SetDirection(), and SetSampleRate().

7.4.4.11 GDT clArraySensor::fShadeCoeff [protected]

Shading coefficient.

Definition at line 51 of file ArraySensor.hh.

Referenced by clArraySensor(), Put(), and SetShading().

7.4.4.12 GDT clArraySensor::fLeftDistance [protected]

Left side sensor distance.

Definition at line 52 of file ArraySensor.hh.

Referenced by Initialize(), and SetSoundSpeed().

7.4.4.13 GDT clArraySensor::fRightDistance [protected]

Right side sensor distance.

Definition at line 53 of file ArraySensor.hh.

Referenced by Initialize(), and SetSoundSpeed().

7.4.4.14 GDT clArraySensor::fLeftTime [protected]

Left side sensor delay.

Definition at line 54 of file ArraySensor.hh.

Referenced by SetDirection(), and SetSoundSpeed().

7.4.4.15 GDT clArraySensor::fRightTime [protected]

Right side sensor delay.

Definition at line 55 of file ArraySensor.hh.

Referenced by SetDirection(), and SetSoundSpeed().

7.4.4.16 GDT clArraySensor::fDirection [protected]

Beam direction for delay.

Definition at line 56 of file ArraySensor.hh.

7.4.4.17 clDSPOp clArraySensor::DSP [protected]

Definition at line 57 of file ArraySensor.hh.

7.4.4.18 clFilter clArraySensor::Filter [protected]

Definition at line 58 of file ArraySensor.hh.

Referenced by Initialize(), Put(), and SetDirection().

7.4.4.19 clDSPVector<GDT> clArraySensor::dspvInData [protected]

Definition at line 59 of file ArraySensor.hh.

Referenced by Put().

7.4.4.20 clDSPVector<GDT> clArraySensor::dspvProcData [protected]

Definition at line 60 of file ArraySensor.hh.

Referenced by Initialize(), and Put().

7.4.4.21 clDSPVector<GDT> clArraySensor::dspvOutData [protected]

Definition at line 61 of file ArraySensor.hh.

Referenced by Get(), and Put().

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

- **ArraySensor.hh**
- **ArraySensor.cc**

7.5 clAudio Class Reference

Class for OSS audio IO operations.

`#include <Audio.hh>`

Public Member Functions

- **clAudio** ()
- **clAudio** (const char ∗, int ∗, int ∗, int ∗, int)
- **~clAudio** ()
- bool **Open** (const char ∗, int ∗, int ∗, int ∗, int)

  Open device.

- void **Close** ()

  Close device.

- bool **SetParams** (int ∗, int ∗, int ∗)

  Set device parameters.

- bool **SetFormat** (int ∗)

  Set format.

- bool **SetChannels** (int ∗)

  Set channel count.

- bool **SetSampleRate** (int ∗)

  Set samplerate.

- int **Read** (void ∗, int)

  Read data to buffer.

- int **Write** (const void ∗, int)

  Write data from buffer.

- bool **Sync** ()

  Wait for buffer to be played empty.

- bool **Reset** ()

  Stops any ongoing action and returns 'normal' state (for example stops recording).

- bool **Post** ()

  Indicates driver that there will be pause in IO action.

- int **GetFragmentSize** ()

Get internal buffering fragment size.

- bool **SubDivide** (int ∗)

  Undocumented feature of OSS.

- bool **SetFragment** (int, int)

  Set buffering.

- int **GetFormats** ()

  Get supported formats.

- bool **GetInBufInfo** (audio_buf_info ∗)

  Get input buffer info.

- bool **GetOutBufInfo** (audio_buf_info ∗)

  Get output buffer info.

- bool **GetInBufStat** (count_info ∗)

  Get input buffer status.

- bool **GetOutBufStat** (count_info ∗)

  Get output buffer status.

- int **GetCaps** ()

  Get device capabilities.

- bool **SetDuplex** ()

  Enable full duplex.

- bool **SetNonBlock** ()

  Set always nonblocking io.

- bool **SetSynchronous** ()

  Set synchronous operation.

- int **GetTrigger** ()

  Get trigger mask.

- bool **SetTrigger** (int)

  Set trigger mask.

- bool **MapInBuffer** (buffmem_desc ∗)

  Map input buffer.

- bool **MapOutBuffer** (buffmem_desc ∗)

  Map output buffer.

- int **GetOutDelay** ()

  Get output delay.

- bool **GetErrorInfo** (audio_errinfo ∗)

  Get detailed error information.

- int **GetVersion** ()

  Get OSS version.

- void **DeIntStereo** (const signed short ∗, signed short ∗, signed short ∗, unsigned int)

  Methods for (de)interleaving stereo data.

- void **DeIntStereo** (const unsigned char ∗, unsigned char ∗, unsigned char ∗, unsigned int)
- void **IntStereo** (signed short ∗, const signed short ∗, const signed short ∗, unsigned int)
- void **IntStereo** (unsigned char ∗, const unsigned char ∗, const unsigned char ∗, unsigned int)
- int **GetHandle** ()

  Get device handle.

- int **GetError** ()

  Get errno code.

- int **GetErrno** ()

Private Attributes

- int **iDevH**
- int **iErrorCode**

7.5.1 Detailed Description

Class for OSS audio IO operations.

Definition at line 49 of file Audio.hh.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 clAudio::clAudio ()

Definition at line 43 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.2.2 clAudio::clAudio (const char *, int *, int *, int *, int)

Definition at line 50 of file Audio.cc.

References iDevH, iErrorCode, and Open().

7.5.2.3 clAudio::~clAudio ()

Definition at line 59 of file Audio.cc.

References Close().

7.5.3 Member Function Documentation

7.5.3.1 bool clAudio::Open (const char *, int *, int *, int *, int)

Open device.

Note:
Check format, channels and samplerate for changes!

Parameters:
cpDevName Device name

ipFormat Format (SND_FMT_*)

ipSampleRate Samplerate

ipChannels Channel count

iType Open mode, read, write or full-duplex

Returns:
Success

Definition at line 65 of file Audio.cc.

References AUDIO_DUPLEX, AUDIO_READ, AUDIO_WRITE, iDevH, and i-ErrorCode.

Referenced by clSoundSrv2::AudioInThread(), clAudio(), clSoundUI::SoundIn-Thread(), and clSoundUI::SoundOutThread().

7.5.3.2 void clAudio::Close ()

Close device.

Definition at line 118 of file Audio.cc.

References iDevH.

Referenced by clBeamAudioUI::AudioOutThread(), clSoundUI::SoundIn-Thread(), clSoundUI::SoundOutThread(), and ~clAudio().

7.5.3.3 bool clAudio::SetParams (int *, int *, int *)

Set device parameters.

Parameters:

ipFormat Format (SND_FMT_*)

ipChannels Channel count

ipSampleRate Samplerate

Returns:

Success

Definition at line 125 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.4 bool clAudio::SetFormat (int *)

Set format.

Parameters:

ipFormat Format (SND_FMT_*)

Returns:

Success

Definition at line 147 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.5 bool clAudio::SetChannels (int *)

Set channel count.

Parameters:

ipChannels Channel count

Returns:

Success

Definition at line 160 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.6 bool clAudio::SetSampleRate (int *)

Set samplerate.

Parameters:
ipSampleRate Samplerate

Returns:
Success

Definition at line 172 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.7 int clAudio::Read (void *, int)

Read data to buffer.

Parameters:
vpData Buffer

iBytes Number of bytes to read

Returns:
Number of bytes read

Definition at line 184 of file Audio.cc.

References iDevH, and iErrorCode.

Referenced by clSoundSrv2::AudioInThread().

7.5.3.8 int clAudio::Write (const void *, int)

Write data from buffer.

Parameters:
vpData Buffer

iBytes Number of bytes to write

Returns:
Number of bytes written

Definition at line 206 of file Audio.cc.

References iDevH, and iErrorCode.

Referenced by clBeamAudioUI::AudioOutThread(), main(), clSoundUI::SoundInThread(), and clSoundUI::SoundOutThread().

7.5.3.9 bool clAudio::Sync ()

Wait for buffer to be played empty.

Returns:
Success

Definition at line 228 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.10 bool clAudio::Reset ()

Stops any ongoing action and returns 'normal' state (for example stops recording).

Returns:
Success

Definition at line 240 of file Audio.cc.

References iDevH, and iErrorCode.

Referenced by clSoundUI::SoundInThread(), and clSoundUI::SoundOutThread().

7.5.3.11 bool clAudio::Post ()

Indicates driver that there will be pause in IO action.

Returns:
Success

Definition at line 252 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.12 int clAudio::GetFragmentSize ()

Get internal buffering fragment size.

Returns:
Internal fragment size in bytes

Definition at line 264 of file Audio.cc.

References iDevH, and iErrorCode.

Referenced by clSoundSrv2::AudioInThread(), and clSoundSrv::clSoundSrv().

7.5.3.13 bool clAudio::SubDivide (int *)

Undocumented feature of OSS.

Definition at line 278 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.14 bool clAudio::SetFragment (int, int)

Set buffering.

Parameters:

iFragSize Fragment size x where fragsize = 2^x

iFragCount Number of fragments, 7fff = no limit

Definition at line 290 of file Audio.cc.

References iDevH, and iErrorCode.

Referenced by clSoundSrv2::AudioInThread().

7.5.3.15 int clAudio::GetFormats ()

Get supported formats.

Returns:

Formats, 0 on error

Definition at line 307 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.16 bool clAudio::GetInBufInfo (audio_buf_info *)

Get input buffer info.

Parameters:

spInBufInfo Input buffer info (audio_buf_info struct)

Returns:

Success

Definition at line 321 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.17 bool clAudio::GetOutBufInfo (audio_buf_info *)

Get output buffer info.

Parameters:

spOutBufInfo Output buffer info (audio_buf_info struct)

Returns:

Success

Definition at line 333 of file Audio.cc.

References iDevH, and iErrorCode.

Referenced by clBeamAudioUI::AudioOutThread(), main(), and clSound-UI::SoundInThread().

7.5.3.18 bool clAudio::GetInBufStat (count_info *)

Get input buffer status.

Parameters:

spInBufStat Input buffer status (count_info struct)

Returns:

Success

Definition at line 345 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.19 bool clAudio::GetOutBufStat (count_info *)

Get output buffer status.

Parameters:

spOutBufStat Output buffer status (count_info struct)

Returns:

Success

Definition at line 357 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.20 int clAudio::GetCaps ()

Get device capabilities.

Returns:

DSP_CAP_* bitmask, 0 on error

Definition at line 369 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.21 bool clAudio::SetDuplex ()

Enable full duplex.

Returns:
Success

Definition at line 383 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.22 bool clAudio::SetNonBlock ()

Set always nonblocking io.

Returns:
Success

Definition at line 395 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.23 bool clAudio::SetSynchronous ()

Set synchronous operation.

Returns:
Success

Definition at line 407 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.24 int clAudio::GetTrigger ()

Get trigger mask.

Returns:
Trigger mask, 0 on error

Definition at line 419 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.25 bool clAudio::SetTrigger (int)

Set trigger mask.

Parameters:
iTrigger Trigger mask

Returns:
Success

Definition at line 433 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.26 bool clAudio::MapInBuffer (buffmem_desc *)

Map input buffer.

Parameters:
spBufferInfo Mapping info (buffmem_desc struct)

Returns:
Success

Definition at line 445 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.27 bool clAudio::MapOutBuffer (buffmem_desc *)

Map output buffer.

Parameters:
spBufferInfo Mapping info (buffmem_desc struct)

Returns:
Success

Definition at line 457 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.28 int clAudio::GetOutDelay ()

Get output delay.

Returns:
Output delay (samples)

Definition at line 469 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.29 bool clAudio::GetErrorInfo (audio_errinfo ∗)

Get detailed error information.

Note:
Commercial OSS driver only!

Parameters:
spErrorInfo Detailed error information

Returns:
Success

Definition at line 484 of file Audio.cc.

References iDevH, and iErrorCode.

7.5.3.30 int clAudio::GetVersion ()

Get OSS version.

Returns:
OSS version, 0 on error

Definition at line 497 of file Audio.cc.

References iDevH, and iErrorCode.

Referenced by clSoundSrv2::AudioInThread(), and clSoundSrv::clSoundSrv().

7.5.3.31 void clAudio::DeIntStereo (const signed short ∗, signed short ∗, signed short ∗, unsigned *int*)

Methods for (de)interleaving stereo data.

Definition at line 511 of file Audio.cc.

7.5.3.32 void clAudio::DeIntStereo (const unsigned char ∗, unsigned char ∗, unsigned char ∗, unsigned *int*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 525 of file Audio.cc.

7.5.3.33 void clAudio::IntStereo (signed short ∗, const signed short ∗, const signed short ∗, unsigned *int*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 538 of file Audio.cc.

7.5.3.34 void clAudio::IntStereo (unsigned char *, const unsigned char *, const unsigned char *, unsigned *int*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 552 of file Audio.cc.

7.5.3.35 int clAudio::GetHandle () `[inline]`

Get device handle.

Returns:
File descriptor to device

Definition at line 283 of file Audio.hh.

References iDevH.

Referenced by clSoundSrv::InputExec().

7.5.3.36 int clAudio::GetError () `[inline]`

Get errno code.

Returns:
errno

Definition at line 289 of file Audio.hh.

References iErrorCode.

Referenced by clSoundSrv::clSoundSrv(), main(), and clSoundUI::SoundOutThread().

7.5.3.37 int clAudio::GetErrno () `[inline]`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 291 of file Audio.hh.

References iErrorCode.

7.5.4 Member Data Documentation

7.5.4.1 int clAudio::iDevH `[private]`

Definition at line 51 of file Audio.hh.

Referenced by clAudio(), Close(), GetCaps(), GetErrorInfo(), GetFormats(), GetFragmentSize(), GetHandle(), GetInBufInfo(), GetInBufStat(), GetOutBufInfo(), GetOutBufStat(), GetOutDelay(), GetTrigger(), GetVersion(), MapInBuffer(), MapOutBuffer(), Open(), Post(), Read(), Reset(), SetChannels(), SetDuplex(), SetFormat(), SetFragment(), SetNonBlock(), SetParams(), SetSampleRate(), SetSynchronous(), SetTrigger(), SubDivide(), Sync(), and Write().

7.5.4.2 int clAudio::iErrorCode `[private]`

Definition at line 52 of file Audio.hh.

Referenced by clAudio(), GetCaps(), GetErrno(), GetError(), GetErrorInfo(), GetFormats(), GetFragmentSize(), GetInBufInfo(), GetInBufStat(), GetOutBufInfo(), GetOutBufStat(), GetOutDelay(), GetTrigger(), GetVersion(), MapInBuffer(), MapOutBuffer(), Open(), Post(), Read(), Reset(), SetChannels(), SetDuplex(), SetFormat(), SetFragment(), SetNonBlock(), SetParams(), SetSampleRate(), SetSynchronous(), SetTrigger(), SubDivide(), Sync(), and Write().

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

- **Audio.hh**
- **Audio.cc**

7.6 clAudio3D Class Reference

3D audio engine

`#include <Audio3D.hh>`

Public Member Functions

- **clAudio3D** ()
- ~**clAudio3D** ()
- void **Initialize** (long)

  Initialize.

- void **Uninitialize** ()

  Uninitialize.

- void **SetAngles** (int, int, GDT)

  Set sound direction.

- void **Process** (GDT ∗)

  Process interleaved data.

- void **Process** (GDT ∗, GDT ∗)

  Process noninterleaved data.

- long **GetWindowSize** ()

  Get size of processing window.

Private Member Functions

- void **CopyToLocal** (GDT ∗, const float ∗)
- void **HeadingElevN40** (int)
- void **HeadingElevN30** (int)
- void **HeadingElevN20** (int)
- void **HeadingElevN10** (int)
- void **HeadingElev0** (int)
- void **HeadingElev10** (int)
- void **HeadingElev20** (int)
- void **HeadingElev30** (int)
- void **HeadingElev40** (int)
- void **HeadingElev50** (int)
- void **HeadingElev60** (int)
- void **HeadingElev70** (int)
- void **HeadingElev80** (int)
- void **HeadingElev90** ()
- void **Prepare** (bool)
- void **ProcessDistance** (GDT ∗, GDT ∗, long)

Private Attributes

- bool **bInitialized**
- long **lWindowSize**
- long **lFFTSize**
- long **lSpectSize**
- GDT **fDistance**
- GDT **fAmp**
- GDT * **fpLeftFilt**
- GDT * **fpRightFilt**
- GCDT * **spLeftFilt**
- GCDT * **spRightFilt**
- clAlloc **LeftData**
- clAlloc **RightData**
- clAlloc **LeftFilt**
- clAlloc **RightFilt**
- clAlloc **CLeftFilt**
- clAlloc **CRightFilt**
- clDSPOp **DSP**
- clFilter **FilterLeft**
- clFilter **FilterRight**

7.6.1 Detailed Description

3D audio engine

Definition at line 38 of file Audio3D.hh.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 clAudio3D::clAudio3D ()

Definition at line 298 of file Audio3D.cc.

References bInitialized.

7.6.2.2 clAudio3D::~clAudio3D ()

Definition at line 304 of file Audio3D.cc.

References bInitialized, and Uninitialize().

7.6.3 Member Function Documentation

7.6.3.1 void clAudio3D::CopyToLocal (GDT *, const float *) `[private]`

Definition at line 27 of file Audio3D.cc.

References GDT, and lFFTSize.

Referenced by HeadingElev0(), HeadingElev90(), and Initialize().

7.6.3.2 void clAudio3D::HeadingElevN40 (int) [inline, private]

Definition at line 60 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.3 void clAudio3D::HeadingElevN30 (int) [inline, private]

Definition at line 61 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.4 void clAudio3D::HeadingElevN20 (int) [inline, private]

Definition at line 62 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.5 void clAudio3D::HeadingElevN10 (int) [inline, private]

Definition at line 63 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.6 void clAudio3D::HeadingElev0 (int) [private]

Definition at line 42 of file Audio3D.cc.

References CopyToLocal(), fpLeftFilt, fpRightFilt, and Prepare().

Referenced by SetAngles().

7.6.3.7 void clAudio3D::HeadingElev10 (int) [inline, private]

Definition at line 65 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.8 void clAudio3D::HeadingElev20 (int) [inline, private]

Definition at line 66 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.9 void clAudio3D::HeadingElev30 (int) [inline, private]

Definition at line 67 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.10 void clAudio3D::HeadingElev40 (int) [inline, private]

Definition at line 68 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.11 void clAudio3D::HeadingElev50 (int) [inline, private]

Definition at line 69 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.12 void clAudio3D::HeadingElev60 (int) [inline, private]

Definition at line 70 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.13 void clAudio3D::HeadingElev70 (int) [inline, private]

Definition at line 71 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.14 void clAudio3D::HeadingElev80 (int) [inline, private]

Definition at line 72 of file Audio3D.hh.

Referenced by SetAngles().

7.6.3.15 void clAudio3D::HeadingElev90 () [private]

Definition at line 238 of file Audio3D.cc.

References CopyToLocal(), fpLeftFilt, fpRightFilt, and Prepare().

Referenced by SetAngles().

7.6.3.16 void clAudio3D::Prepare (bool) [private]

Definition at line 246 of file Audio3D.cc.

References DSP, fAmp, fpLeftFilt, fpRightFilt, GDT, lSpectSize, spLeftFilt, and spRightFilt.

Referenced by HeadingElev0(), HeadingElev90(), and Initialize().

7.6.3.17 void clAudio3D::ProcessDistance (GDT *, GDT *, long) [private]

Definition at line 283 of file Audio3D.cc.

References fDistance, and GDT.

Referenced by Process().

7.6.3.18 void clAudio3D::Initialize (long)

Initialize.

Note:
Reinitialization without **Uninitialize()**(p. 63) is allowed

Parameters:
lSize Size of window

Definition at line 310 of file Audio3D.cc.

References bInitialized, CLeftFilt, CopyToLocal(), CRightFilt, DSP, FilterLeft, FilterRight, fpLeftFilt, fpRightFilt, GCDT, GDT, LeftData, LeftFilt, lFFTSize, lSpectSize, lWindowSize, Prepare(), RightData, RightFilt, spLeftFilt, spRightFilt, and Uninitialize().

Referenced by clBeamAudioUI::AllocateBuffers().

7.6.3.19 void clAudio3D::Uninitialize ()

Uninitialize.

Definition at line 334 of file Audio3D.cc.

References bInitialized, DSP, FilterLeft, and FilterRight.

Referenced by Initialize(), and ~clAudio3D().

7.6.3.20 void clAudio3D::SetAngles (int, int, GDT)

Set sound direction.

Note:
Heading and pitch in degrees, distance 0 - 1

Parameters:
iHeading Heading (deg)
iPitch Pitch (deg)

fDist Distance

Definition at line 343 of file Audio3D.cc.

References fDistance, HeadingElev0(), HeadingElev10(), HeadingElev20(), HeadingElev30(), HeadingElev40(), HeadingElev50(), HeadingElev60(), HeadingElev70(), HeadingElev80(), HeadingElev90(), HeadingElevN10(), HeadingElevN20(), HeadingElevN30(), and HeadingElevN40().

Referenced by clBeamAudioUI::OnGdkInput().

7.6.3.21 void clAudio3D::Process (GDT ∗)

Process interleaved data.

Parameters:

fpData New data

Definition at line 363 of file Audio3D.cc.

References DSP, fAmp, FilterLeft, FilterRight, GDT, LeftData, lWindowSize, ProcessDistance(), RightData, spLeftFilt, and spRightFilt.

Referenced by clBeamAudioUI::Process3D().

7.6.3.22 void clAudio3D::Process (GDT ∗, GDT ∗)

Process noninterleaved data.

Parameters:

fpLeftData Left channel data

fpRightData Right channel data

Definition at line 396 of file Audio3D.cc.

References DSP, fAmp, FilterLeft, FilterRight, lWindowSize, ProcessDistance(), spLeftFilt, and spRightFilt.

7.6.3.23 long clAudio3D::GetWindowSize () `[inline]`

Get size of processing window.

Returns:

Size of processing window

Definition at line 119 of file Audio3D.hh.

References lWindowSize.

Referenced by clBeamAudioUI::Process3D().

7.6.4 Member Data Documentation

7.6.4.1 bool clAudio3D::bInitialized [private]

Definition at line 40 of file Audio3D.hh.

Referenced by clAudio3D(), Initialize(), Uninitialize(), and ∼clAudio3D().

7.6.4.2 long clAudio3D::lWindowSize [private]

Definition at line 41 of file Audio3D.hh.

Referenced by GetWindowSize(), Initialize(), and Process().

7.6.4.3 long clAudio3D::lFFTSize [private]

Definition at line 42 of file Audio3D.hh.

Referenced by CopyToLocal(), and Initialize().

7.6.4.4 long clAudio3D::lSpectSize [private]

Definition at line 43 of file Audio3D.hh.

Referenced by Initialize(), and Prepare().

7.6.4.5 GDT clAudio3D::fDistance [private]

Definition at line 44 of file Audio3D.hh.

Referenced by ProcessDistance(), and SetAngles().

7.6.4.6 GDT clAudio3D::fAmp [private]

Definition at line 45 of file Audio3D.hh.

Referenced by Prepare(), and Process().

7.6.4.7 GDT∗ clAudio3D::fpLeftFilt [private]

Definition at line 46 of file Audio3D.hh.

Referenced by HeadingElev0(), HeadingElev90(), Initialize(), and Prepare().

7.6.4.8 GDT∗ clAudio3D::fpRightFilt [private]

Definition at line 47 of file Audio3D.hh.

Referenced by HeadingElev0(), HeadingElev90(), Initialize(), and Prepare().

7.6.4.9 GCDT* clAudio3D::spLeftFilt [private]

Definition at line 48 of file Audio3D.hh.

Referenced by Initialize(), Prepare(), and Process().

7.6.4.10 GCDT* clAudio3D::spRightFilt [private]

Definition at line 49 of file Audio3D.hh.

Referenced by Initialize(), Prepare(), and Process().

7.6.4.11 clAlloc clAudio3D::LeftData [private]

Definition at line 50 of file Audio3D.hh.

Referenced by Initialize(), and Process().

7.6.4.12 clAlloc clAudio3D::RightData [private]

Definition at line 51 of file Audio3D.hh.

Referenced by Initialize(), and Process().

7.6.4.13 clAlloc clAudio3D::LeftFilt [private]

Definition at line 52 of file Audio3D.hh.

Referenced by Initialize().

7.6.4.14 clAlloc clAudio3D::RightFilt [private]

Definition at line 53 of file Audio3D.hh.

Referenced by Initialize().

7.6.4.15 clAlloc clAudio3D::CLeftFilt [private]

Definition at line 54 of file Audio3D.hh.

Referenced by Initialize().

7.6.4.16 clAlloc clAudio3D::CRightFilt [private]

Definition at line 55 of file Audio3D.hh.

Referenced by Initialize().

7.6.4.17 clDSPOp clAudio3D::DSP [private]

Definition at line 56 of file Audio3D.hh.

Referenced by Initialize(), Prepare(), Process(), and Uninitialize().

7.6.4.18 clFilter clAudio3D::FilterLeft [private]

Definition at line 57 of file Audio3D.hh.

Referenced by Initialize(), Process(), and Uninitialize().

7.6.4.19 clFilter clAudio3D::FilterRight [private]

Definition at line 58 of file Audio3D.hh.

Referenced by Initialize(), Process(), and Uninitialize().

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

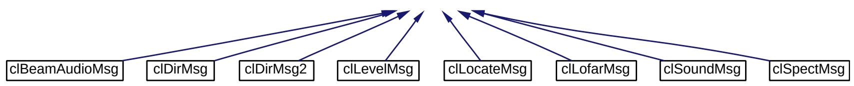
- **Audio3D.hh**
- **Audio3D.cc**

7.7 clBaseMsg Class Reference

Base class for all message handling.

#include <Messages.hh>

Inheritance diagram for clBaseMsg:



Protected Member Functions

- void **EndianConv** (float ∗, const float ∗)

  Result endian conversion.

- void **EndianConv** (double ∗, const double ∗)

7.7.1 Detailed Description

Base class for all message handling.

For communication between userinterfaces and processing servers the usual rule is to first send request message. This is null-terminated string of length GLOBAL_HEADER_LEN. For results there is GLOBAL_HEADER_LEN amount of null-terminated string header data and then endianess converted result data follows.

Set/GetRequest generates/parses this string from request struct. Set/GetResult generates/parses the result message.

Definition at line 437 of file Messages.hh.

7.7.2 Member Function Documentation

7.7.2.1 void clBaseMsg::EndianConv (float ∗, const float ∗) [inline, protected]

Result endian conversion.

Parameters:

fpDest Destination

fpSrc Source

Definition at line 35 of file Messages.cc.

Referenced by clSoundMsg::GetData(), clLocateMsg::GetResult(), clBeamAudioMsg::GetResult(), clLofarMsg::GetResult(), clDirMsg2::GetResult(), clDirMsg::GetResult(), clSpectMsg::GetResult(), clSoundMsg::SetData(), clLocateMsg::SetResult(), clBeamAudioMsg::SetResult(), clLofarMsg::SetResult(), clDirMsg2::SetResult(), clDirMsg::SetResult(), and clSpectMsg::SetResult().

7.7.2.2 void clBaseMsg::EndianConv (double *, const double *) [inline, protected]

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 55 of file Messages.cc.

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

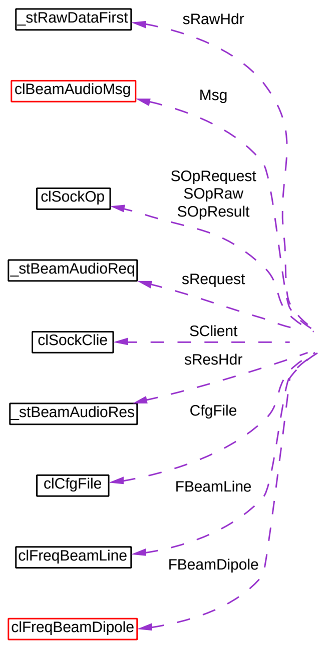
- **Messages.hh**
- **Messages.cc**

7.8 clBeamAudio Class Reference

Server for beamformed audio.

`#include <BeamAudio.hh>`

Collaboration diagram for clBeamAudio:



Public Member Functions

- **clBeamAudio** (int, int)
- ~**clBeamAudio** ()
- int **Main** ()
- void **Stop** ()

Private Member Functions

- bool **GetCfg** ()
- bool **GetRq** ()
- bool **ConnectStream** ()
- bool **InitBeam** ()
- bool **SendFirst** ()
- void **SetDirection** ()
- void **ProcessLoop** ()
- bool **SendResult** (GDT ∗, long)

Private Attributes

- bool **bRun**
- int **iArrayType**
- int **iShadingType**
- long **lStartCh**
- long **lWindowSize**
- long **lSensorCount**
- GDT **fSensorSpacing**
- char **cpStreamSocket** [_POSIX_PATH_MAX+1]
- GDT **fAttenuation**
- **stBeamAudioReq sRequest**
- **stBeamAudioRes sResHdr**
- **stRawDataFirst sRawHdr**
- **clCfgFile CfgFile**
- **clBeamAudioMsg Msg**
- clDSPOp **DSP**
- **clSockClie SClient**
- **clSockOp SOpRaw**
- **clSockOp SOpRequest**
- **clSockOp SOpResult**
- **clFreqBeamDipole FBeamDipole**
- **clFreqBeamLine FBeamLine**

7.8.1 Detailed Description

Server for beamformed audio.

Definition at line 55 of file BeamAudio.hh.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 clBeamAudio::clBeamAudio (int, int)

Definition at line 302 of file BeamAudio.cc.

References bRun, clSockOp::DisableNagle(), fAttenuation, GDT, clSockOp::SetHandle(), SOpRequest, and SOpResult.

7.8.2.2 clBeamAudio::~clBeamAudio ()

Definition at line 312 of file BeamAudio.cc.

7.8.3 Member Function Documentation

7.8.3.1 bool clBeamAudio::GetCfg () [private]

Definition at line 97 of file BeamAudio.cc.

References BA_CFGFILE, BA_DEF_FRAGMENT_SIZE, CfgFile, cpStreamSocket, fSensorSpacing, clCfgFile::GetFlt(), clCfgFile::GetInt(), clCfgFile::GetStr(), iArrayType, iShadingType, lSensorCount, lStartCh, lWindowSize, SD_CFGFILE, and clCfgFile::SetFileName().

Referenced by Main().

7.8.3.2 bool clBeamAudio::GetRq () [private]

Definition at line 113 of file BeamAudio.cc.

References BA_1STREQ_TIMEOUT, clBeamAudioMsg::GetRequest(), GLOBAL_HEADER_LEN, Msg, clSockOp::ReadN(), clSockOp::ReadSelect(), SOpRequest, and sRequest.

Referenced by Main(), and ProcessLoop().

7.8.3.3 bool clBeamAudio::ConnectStream () [private]

Definition at line 125 of file BeamAudio.cc.

References BA_RAW1ST_TIMEOUT, clSockClie::Connect(), cpStreamSocket, _stRawDataReq::iChannel, clSockOp::ReadN(), clSockOp::ReadSelect(), SClient, clSockOp::SetHandle(), SOpRaw, sRawHdr, and clSockOp::WriteN().

Referenced by Main().

7.8.3.4 bool clBeamAudio::InitBeam () [private]

Definition at line 143 of file BeamAudio.cc.

References BA_ARRAY_TYPE_CYLINDER, BA_ARRAY_TYPE_DIPOLE, BA_ARRAY_TYPE_LINE, BA_ARRAY_TYPE_PLANE, BA_ARRAY_TYPE_SPHERE, BA_ARRAY_TYPE_TRIANGLE, bDebug, _stBeamAudioReq::bHighFreq, _stRawDataFirst::dSampleRate, FBeamDipole, FBeamLine, _stBeamAudioReq::fDirection, fSensorSpacing, _stBeamAudioReq::fSoundSpeed, GDT, iArrayType, clFreqBeamLine::Initialize(), clFreqBeamDipole::Initialize(), lSensorCount, lWindowSize, clFreqBeamLine::SetDebug(), clFreqBeamDipole::SetDebug(), clFreqBeamLine::SetDirection(), clFreqBeamDipole::SetDirection(), clFreqBeamLine::SetSoundSpeed(), clFreqBeamDipole::SetSoundSpeed(), sRawHdr, and sRequest.

Referenced by Main().

7.8.3.5 bool clBeamAudio::SendFirst () [private]

Definition at line 190 of file BeamAudio.cc.

References _stRawDataFirst::dSampleRate, GLOBAL_HEADER_LEN, _stBeamAudioFirst::iSampleRate, _stBeamAudioFirst::lBufLength, lWindowSize, Msg, clBeamAudioMsg::SetFirst(), SOpResult, sRawHdr, and clSockOp::WriteN().

Referenced by Main().

7.8.3.6 void clBeamAudio::SetDirection () [private]

Definition at line 204 of file BeamAudio.cc.

References BA_ARRAY_TYPE_CYLINDER, BA_ARRAY_TYPE_DIPOLE, BA_ARRAY_TYPE_LINE, BA_ARRAY_TYPE_PLANE, BA_ARRAY_TYPE_SPHERE, BA_ARRAY_TYPE_TRIANGLE, _stBeamAudioReq::bHighFreq, FBeamDipole, FBeamLine, _stBeamAudioReq::fDirection, iArrayType, clFreqBeamLine::SetDirection(), clFreqBeamDipole::SetDirection(), and sRequest.

Referenced by ProcessLoop().

7.8.3.7 void clBeamAudio::ProcessLoop () [private]

Definition at line 226 of file BeamAudio.cc.

References BA_ARRAY_TYPE_CYLINDER, BA_ARRAY_TYPE_DIPOLE, BA_ARRAY_TYPE_LINE, BA_ARRAY_TYPE_PLANE, BA_ARRAY_TYPE_SPHERE, BA_ARRAY_TYPE_TRIANGLE, bRun, DSP, FBeamDipole, FBeamLine, GDT, clFreqBeamLine::Get(), clFreqBeamDipole::Get(), GetRq(), iArrayType, _stRawDataFirst::iChannels, lStartCh, lWindowSize, clFreqBeamLine::Put(), clFreqBeamDipole::Put(), clSockOp::ReadN(), clSockOp::ReadSelect(), SendResult(), SetDirection(), SOpRaw, SOpRequest, and sRawHdr.

Referenced by Main().

7.8.3.8 bool clBeamAudio::SendResult (GDT *, long) [private]

Definition at line 281 of file BeamAudio.cc.

References DSP, fAttenuation, _stBeamAudioReq::fDirection, _stBeamAudioRes::fDirection, _stBeamAudioRes::fPeakLevel, GDT, GLOBAL_HEADER_LEN, _stBeamAudioRes::lBufLength, Msg, clBeamAudioMsg::SetResult(), SOpResult, sRequest, sResHdr, _stBeamAudioRes::sTimeStamp, and clSockOp::WriteN().

Referenced by ProcessLoop().

7.8.3.9 int clBeamAudio::Main ()

Definition at line 317 of file BeamAudio.cc.

References bDebug, ConnectStream(), GetCfg(), GetRq(), InitBeam(), ProcessLoop(), and SendFirst().

Referenced by main().

7.8.3.10 void clBeamAudio::Stop () [inline]

Definition at line 92 of file BeamAudio.hh.

References bRun.

7.8.4 Member Data Documentation

7.8.4.1 bool clBeamAudio::bRun [private]

Definition at line 57 of file BeamAudio.hh.

Referenced by clBeamAudio(), ProcessLoop(), and Stop().

7.8.4.2 int clBeamAudio::iArrayType [private]

Definition at line 59 of file BeamAudio.hh.

Referenced by GetCfg(), InitBeam(), ProcessLoop(), and SetDirection().

7.8.4.3 int clBeamAudio::iShadingType [private]

Definition at line 60 of file BeamAudio.hh.

Referenced by GetCfg().

7.8.4.4 long clBeamAudio::lStartCh [private]

Definition at line 61 of file BeamAudio.hh.

Referenced by GetCfg(), and ProcessLoop().

7.8.4.5 long clBeamAudio::lWindowSize [private]

Definition at line 62 of file BeamAudio.hh.

Referenced by GetCfg(), InitBeam(), ProcessLoop(), and SendFirst().

7.8.4.6 long clBeamAudio::lSensorCount [private]

Definition at line 63 of file BeamAudio.hh.

Referenced by GetCfg(), and InitBeam().

7.8.4.7 GDT clBeamAudio::fSensorSpacing [private]

Definition at line 64 of file BeamAudio.hh.

Referenced by GetCfg(), and InitBeam().

7.8.4.8 char clBeamAudio::cpStreamSocket[_POSIX_PATH_MAX + 1] [private]

Definition at line 65 of file BeamAudio.hh.

Referenced by ConnectStream(), and GetCfg().

7.8.4.9 GDT clBeamAudio::fAttenuation [private]

Definition at line 67 of file BeamAudio.hh.

Referenced by clBeamAudio(), and SendResult().

7.8.4.10 stBeamAudioReq clBeamAudio::sRequest [private]

Definition at line 68 of file BeamAudio.hh.

Referenced by GetRq(), InitBeam(), SendResult(), and SetDirection().

7.8.4.11 stBeamAudioRes clBeamAudio::sResHdr [private]

Definition at line 69 of file BeamAudio.hh.

Referenced by SendResult().

7.8.4.12 stRawDataFirst clBeamAudio::sRawHdr [private]

Definition at line 70 of file BeamAudio.hh.

Referenced by ConnectStream(), InitBeam(), ProcessLoop(), and SendFirst().

7.8.4.13 clCfgFile clBeamAudio::CfgFile [private]

Definition at line 71 of file BeamAudio.hh.

Referenced by GetCfg().

7.8.4.14 clBeamAudioMsg clBeamAudio::Msg `[private]`

Definition at line 72 of file BeamAudio.hh.

Referenced by GetRq(), SendFirst(), and SendResult().

7.8.4.15 clDSPOp clBeamAudio::DSP `[private]`

Definition at line 73 of file BeamAudio.hh.

Referenced by ProcessLoop(), and SendResult().

7.8.4.16 clSockClie clBeamAudio::SClient `[private]`

Definition at line 74 of file BeamAudio.hh.

Referenced by ConnectStream().

7.8.4.17 clSockOp clBeamAudio::SOpRaw `[private]`

Definition at line 75 of file BeamAudio.hh.

Referenced by ConnectStream(), and ProcessLoop().

7.8.4.18 clSockOp clBeamAudio::SOpRequest `[private]`

Definition at line 76 of file BeamAudio.hh.

Referenced by clBeamAudio(), GetRq(), and ProcessLoop().

7.8.4.19 clSockOp clBeamAudio::SOpResult `[private]`

Definition at line 77 of file BeamAudio.hh.

Referenced by clBeamAudio(), SendFirst(), and SendResult().

7.8.4.20 clFreqBeamDipole clBeamAudio::FBeamDipole `[private]`

Definition at line 78 of file BeamAudio.hh.

Referenced by InitBeam(), ProcessLoop(), and SetDirection().

7.8.4.21 clFreqBeamLine clBeamAudio::FBeamLine `[private]`

Definition at line 79 of file BeamAudio.hh.

Referenced by InitBeam(), ProcessLoop(), and SetDirection().

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

- **BeamAudio.hh**
- **BeamAudio.cc**

7.9 _stBeamAudioFirst Struct Reference

BeamAudio: result header.

`#include <Messages.hh>`

Public Attributes

- long **lBufLength**
  Buffer length.
- int **iSampleRate**
  Samplerate.

7.9.1 Detailed Description

BeamAudio: result header.

Definition at line 367 of file Messages.hh.

7.9.2 Member Data Documentation

7.9.2.1 long _stBeamAudioFirst::lBufLength

Buffer length.

Definition at line 369 of file Messages.hh.

Referenced by clBeamAudioUI::AllocateBuffers(), clBeamAudioMsg::GetFirst(), clBeamAudio::SendFirst(), and clBeamAudioMsg::SetFirst().

7.9.2.2 int _stBeamAudioFirst::iSampleRate

Samplerate.

Definition at line 370 of file Messages.hh.

Referenced by clBeamAudioUI::AllocateBuffers(), clBeamAudioMsg::GetFirst(), clBeamAudio::SendFirst(), and clBeamAudioMsg::SetFirst().

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

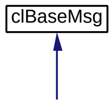
- **Messages.hh**

7.10 clBeamAudioMsg Class Reference

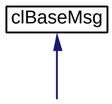
BeamAudio server communication.

```
#include <Messages.hh>
```

Inheritance diagram for clBeamAudioMsg:



Collaboration diagram for clBeamAudioMsg:



Public Member Functions

- void **SetRequest** (char ∗, const **stpBeamAudioReq**)
- void **GetRequest** (const char ∗, **stpBeamAudioReq**)
- void **SetFirst** (char ∗, const **stpBeamAudioFirst**)
- void **GetFirst** (const char ∗, **stpBeamAudioFirst**)
- void **SetResult** (void ∗, const **stpBeamAudioRes**, const float ∗)
- void **SetResult** (void ∗, const **stpBeamAudioRes**, const double ∗)
- void **GetResult** (const void ∗, **stpBeamAudioRes**, float ∗)
- void **GetResult** (const void ∗, **stpBeamAudioRes**, double ∗)

7.10.1 Detailed Description

BeamAudio server communication.

Definition at line 568 of file Messages.hh.

7.10.2 Member Function Documentation

7.10.2.1 void clBeamAudioMsg::SetRequest (char ∗, const *stpBeamAudioReq*)

Definition at line 1056 of file Messages.cc.

Referenced by clBeamAudioUI::SendSettings(), and clBeamAudioUI::Update-Settings().

7.10.2.2 void clBeamAudioMsg::GetRequest (const char *, stpBeamAudioReq)

Definition at line 1076 of file Messages.cc.

References _stBeamAudioReq::bHighFreq, _stBeamAudioReq::fDirection, _stBeamAudioReq::fSoundSpeed, and MSG_TRUE.

Referenced by clBeamAudio::GetRq().

7.10.2.3 void clBeamAudioMsg::SetFirst (char *, const *stpBeamAudioFirst*)

Definition at line 1089 of file Messages.cc.

References GLOBAL_HEADER_LEN, _stBeamAudioFirst::iSampleRate, and _stBeamAudioFirst::lBufLength.

Referenced by clBeamAudio::SendFirst().

7.10.2.4 void clBeamAudioMsg::GetFirst (const char *, stpBeamAudioFirst)

Definition at line 1104 of file Messages.cc.

References _stBeamAudioFirst::iSampleRate, and _stBeamAudioFirst::lBufLength.

Referenced by clBeamAudioUI::SendSettings().

7.10.2.5 void clBeamAudioMsg::SetResult (void *, const *stpBeamAudioRes*, const float *)

Definition at line 1113 of file Messages.cc.

References clBaseMsg::EndianConv(), _stBeamAudioRes::fDirection, _stBeamAudioRes::fPeakLevel, GLOBAL_HEADER_LEN, _stBeamAudioRes::lBufLength, and _stBeamAudioRes::sTimeStamp.

Referenced by clBeamAudio::SendResult().

7.10.2.6 void clBeamAudioMsg::SetResult (void *, const *stpBeamAudioRes*, const double *)

Definition at line 1142 of file Messages.cc.

References clBaseMsg::EndianConv(), _stBeamAudioRes::fDirection, _stBeamAudioRes::fPeakLevel, GLOBAL_HEADER_LEN, _stBeamAudioRes::lBufLength, and _stBeamAudioRes::sTimeStamp.

7.10.2.7 void clBeamAudioMsg::GetResult (const void *, stpBeamAudioRes, float *)

Definition at line 1171 of file Messages.cc.

References clBaseMsg::EndianConv(), _stBeamAudioRes::fDirection, _stBeamAudioRes::fPeakLevel, GLOBAL_HEADER_LEN, _stBeamAudioRes::lBufLength, and _stBeamAudioRes::sTimeStamp.

Referenced by clBeamAudioUI::OnGdkInput().

7.10.2.8 void clBeamAudioMsg::GetResult (const void *, stpBeamAudioRes, double *)

Definition at line 1195 of file Messages.cc.

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

- **Messages.hh**
- **Messages.cc**

7.11 _stBeamAudioReq Struct Reference

BeamAudio: request.

`#include <Messages.hh>`

Public Attributes

- float **fDirection**

  Direction of beam (rad).

- float **fSoundSpeed**

  Speed of sound (m/s).

- bool **bHighFreq**

  High frequency content.

7.11.1 Detailed Description

BeamAudio: request.

Definition at line 357 of file Messages.hh.

7.11.2 Member Data Documentation

7.11.2.1 float _stBeamAudioReq::fDirection

Direction of beam (rad).

Definition at line 359 of file Messages.hh.

Referenced by clBeamAudioMsg::GetRequest(), clBeamAudio::InitBeam(), clBeamAudioUI::OnValueChanged(), clBeamAudio::SendResult(), clBeamAudioUI::SendSettings(), and clBeamAudio::SetDirection().

7.11.2.2 float _stBeamAudioReq::fSoundSpeed

Speed of sound (m/s).

Definition at line 360 of file Messages.hh.

Referenced by clBeamAudioMsg::GetRequest(), clBeamAudio::InitBeam(), and clBeamAudioUI::SendSettings().

7.11.2.3 bool _stBeamAudioReq::bHighFreq

High frequency content.

Definition at line 361 of file Messages.hh.

Referenced by clBeamAudioMsg::GetRequest(), clBeamAudio::InitBeam(), clBeamAudioUI::OnToggled(), clBeamAudioUI::SendSettings(), and clBeamAudio::SetDirection().

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

- **Messages.hh**

7.12 _stBeamAudioRes Struct Reference

BeamAudio: result.

`#include <Messages.hh>`

Public Attributes

- timeval **sTimeStamp**

  Timestamp.

- long **lBufLength**

  Buffer length.

- float **fPeakLevel**

  Peak level (dB).

- float **fDirection**

  Direction (rad).

7.12.1 Detailed Description

BeamAudio: result.

Definition at line 376 of file Messages.hh.

7.12.2 Member Data Documentation

7.12.2.1 struct timeval _stBeamAudioRes::sTimeStamp

Timestamp.

Definition at line 378 of file Messages.hh.

Referenced by clBeamAudioMsg::GetResult(), clBeamAudio::SendResult(), and clBeamAudioMsg::SetResult().

7.12.2.2 long _stBeamAudioRes::lBufLength

Buffer length.

Definition at line 379 of file Messages.hh.

Referenced by clBeamAudioMsg::GetResult(), clBeamAudio::SendResult(), and clBeamAudioMsg::SetResult().

7.12.2.3 float _stBeamAudioRes::fPeakLevel

Peak level (dB).

Definition at line 380 of file Messages.hh.

Referenced by clBeamAudioMsg::GetResult(), clBeamAudioUI::OnGdkInput(), clBeamAudio::SendResult(), and clBeamAudioMsg::SetResult().

7.12.2.4 float _stBeamAudioRes::fDirection

Direction (rad).

Definition at line 381 of file Messages.hh.

Referenced by clBeamAudioMsg::GetResult(), clBeamAudioUI::OnGdkInput(), clBeamAudio::SendResult(), and clBeamAudioMsg::SetResult().

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

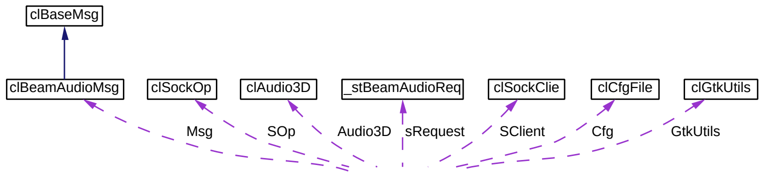
- **Messages.hh**

7.13 clBeamAudioUI Class Reference

User interface for beam audio server.

`#include <BeamAudioUI.hh>`

Collaboration diagram for clBeamAudioUI:



Public Member Functions

- **clBeamAudioUI** (int ∗, char ∗∗∗)
- ∼**clBeamAudioUI** ()
- int **Main** ()
- gboolean **OnDelete** (GtkWidget ∗, GdkEvent ∗, gpointer)
- void **OnConnectClick** (GtkButton ∗, gpointer)
- void **OnValueChanged** (GtkAdjustment ∗, gpointer)
- void **OnToggled** (GtkToggleButton ∗, gpointer)
- void **OnGdkInput** (gpointer, gint, GdkInputCondition)
- void ∗ **AudioOutThread** (void ∗)

Private Member Functions

- void **Build** ()
- void **BuildTable1** ()
- void **BuildTable2** ()
- void **ConnectSignals** ()
- bool **ParseServerStr** (char ∗, int ∗, const char ∗)
- bool **InitConnection** (const char ∗, int)
- bool **SendSettings** ()
- void **AllocateBuffers** (const **stpBeamAudioFirst**)
- bool **InitAudio** (**clAudio** &, clAudioA2 &)
- void **UpdateSettings** ()
- void **ConvertMS** ()
- void **Process3D** ()
- void **Dither** ()
- void **ConvertFromDither** ()

Private Attributes

- volatile bool **bConnected**
- bool **bDither**
- bool **b3DAudio**
- bool **bALSA**
- int **iSampleRate**
- int **iRandH**
- int **iPrevHeading**
- long **lFragSize**
- long **lDataCount**
- long **lMsgSize**
- long **lClips**
- volatile long **lBufIdx**
- GDT * **fpInAudio**
- GDT * **fpSrcAudio**
- BAUI_SND_DATATYPE * **ipOutAudio** [BAUI_AUDIO_BUFCOUNT]
- pthread_t **ptidAudio**
- **stBeamAudioReq sRequest**
- guint **guSbCtxt**
- gint **giGdkTag**
- GList * **glServer**
- GtkWidget * **gwWindow**
- GtkWidget * **gwVBox**
- GtkWidget * **gwTable1**
- GtkWidget * **gwLServer**
- GtkWidget * **gwCServer**
- GtkWidget * **gwBConnect**
- GtkWidget * **gwTable2**
- GtkWidget * **gwLSoundSpeed**
- GtkWidget * **gwESoundSpeed**
- GtkWidget * **gwCBHighFreq**
- GtkWidget * **gwCBDither**
- GtkWidget * **gwCB3DAudio**
- GtkWidget * **gwLDirection**
- GtkObject * **goADirection**
- GtkWidget * **gwHSDirection**
- GtkWidget * **gwStatusBar**
- clAlloc **MessageBuf**
- clAlloc **InAudioBuf**
- clAlloc **SrcAudioBuf**
- clAlloc **DitherBuf**
- clAlloc **DitherRandBuf**
- clAlloc **OutAudioBuf** [BAUI_AUDIO_BUFCOUNT]
- **clAudio3D Audio3D**
- **clCfgFile Cfg**
- clDSPOp **DSP**

- **clBeamAudioMsg Msg**
- clMutex **OutBufMutex** [BAUI_AUDIO_BUFCOUNT]
- **clGtkUtils GtkUtils**
- **clSockClie SClient**
- **clSockOp SOp**

7.13.1 Detailed Description

User interface for beam audio server.

Definition at line 76 of file BeamAudioUI.hh.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 clBeamAudioUI::clBeamAudioUI (int ∗, char ∗∗∗)

Definition at line 600 of file BeamAudioUI.cc.

References b3DAudio, bALSA, BAUI_CFGFILE, BAUI_VER_MAJ, BAUI_VER_MIN, BAUI_VER_PL, bConnected, bDither, Build(), Cfg, clCfgFile::GetInt(), iPrevHeading, iRandH, and clCfgFile::SetFileName().

7.13.2.2 clBeamAudioUI::∼clBeamAudioUI ()

Definition at line 636 of file BeamAudioUI.cc.

References iRandH.

7.13.3 Member Function Documentation

7.13.3.1 void clBeamAudioUI::Build () `[private]`

Definition at line 96 of file BeamAudioUI.cc.

References BAUI_DEF_DIR_RANGE, BAUI_PADDING, BuildTable1(), BuildTable2(), ConnectSignals(), goADirection, guSbCtxt, gwHSDirection, gwLDirection, gwStatusBar, gwVBox, and gwWindow.

Referenced by clBeamAudioUI().

7.13.3.2 void clBeamAudioUI::BuildTable1 () `[private]`

Definition at line 157 of file BeamAudioUI.cc.

References BAUI_HOSTFILE, BAUI_PADDING, BAUI_SERVER_MAXLEN, clGtkUtils::ComboListFromFile(), glServer, GtkUtils, gwBConnect, gwCServer, gwLServer, gwTable1, and gwVBox.

Referenced by Build().

7.13.3.3 void clBeamAudioUI::BuildTable2 () [private]

Definition at line 199 of file BeamAudioUI.cc.

References BAUI_DEF_SOUNDSPEED, BAUI_PADDING, gwCB3DAudio, gwCBDither, gwCBHighFreq, gwESoundSpeed, gwLSoundSpeed, gwTable2, and gwVBox.

Referenced by Build().

7.13.3.4 void clBeamAudioUI::ConnectSignals () [private]

Definition at line 254 of file BeamAudioUI.cc.

References goADirection, gwBConnect, gwCB3DAudio, gwCBDither, gwCBHighFreq, gwWindow, WrapOnConnectClick(), WrapOnDelete(), WrapOnToggled(), and WrapOnValueChanged().

Referenced by Build().

7.13.3.5 bool clBeamAudioUI::ParseServerStr (char *, int *, const char *) [private]

Definition at line 279 of file BeamAudioUI.cc.

References BAUI_SERVER_MAXLEN.

Referenced by OnConnectClick().

7.13.3.6 bool clBeamAudioUI::InitConnection (const char *, int) [private]

Definition at line 302 of file BeamAudioUI.cc.

References BAUI_REQ_PROC, bConnected, clSockClie::Connect(), giGdkTag, GLOBAL_HEADER_LEN, lClips, SClient, SendSettings(), clSockOp::SetHandle(), SOp, WrapOnGdkInput(), and clSockOp::WriteN().

Referenced by OnConnectClick().

7.13.3.7 bool clBeamAudioUI::SendSettings () [private]

Definition at line 334 of file BeamAudioUI.cc.

References AllocateBuffers(), BAUI_1ST_TIMEOUT, _stBeamAudioReq::bHighFreq, _stBeamAudioReq::fDirection, _stBeamAudioReq::fSoundSpeed, clBeamAudioMsg::GetFirst(), GLOBAL_HEADER_LEN, goADirection, gwCBHighFreq, gwESoundSpeed, lBufIdx, Msg, ptidAudio, clSockOp::ReadN(), clSockOp::ReadSelect(), clBeamAudioMsg::SetRequest(), SOp, sRequest, WrapAudioOutThread(), and clSockOp::WriteN().

Referenced by InitConnection().

7.13.3.8 void clBeamAudioUI::AllocateBuffers (const *stpBeamAudioFirst*) [private]

Definition at line 360 of file BeamAudioUI.cc.

References Audio3D, BAUI_AUDIO_BUFCOUNT, BAUI_INT_DATATYPE, BAUI_SND_DATATYPE, DitherBuf, DitherRandBuf, fpInAudio, fpSrcAudio, GDT, GLOBAL_HEADER_LEN, InAudioBuf, clAudio3D::Initialize(), ipOutAudio, iRandH, _stBeamAudioFirst::iSampleRate, iSampleRate, _stBeamAudioFirst::lBufLength, lDataCount, lFragSize, lMsgSize, MessageBuf, OutAudioBuf, and SrcAudioBuf.

Referenced by SendSettings().

7.13.3.9 bool clBeamAudioUI::InitAudio (clAudio &, clAudioA2 &) [private]

Definition at line 415 of file BeamAudioUI.cc.

References AUDIO_WRITE, BAUI_SND_BITS, BAUI_SND_DATATYPE, and BAUI_SND_FORMAT.

Referenced by AudioOutThread().

7.13.3.10 void clBeamAudioUI::UpdateSettings () [private]

Definition at line 508 of file BeamAudioUI.cc.

References bConnected, clSockOp::GetErrno(), GLOBAL_HEADER_LEN, Msg, clBeamAudioMsg::SetRequest(), SOp, sRequest, clSockOp::WriteN(), and clSockOp::WriteSelect().

Referenced by OnToggled(), and OnValueChanged().

7.13.3.11 void clBeamAudioUI::ConvertMS () [private]

Definition at line 531 of file BeamAudioUI.cc.

References fpInAudio, fpSrcAudio, and lFragSize.

Referenced by OnGdkInput().

7.13.3.12 void clBeamAudioUI::Process3D () [private]

Definition at line 545 of file BeamAudioUI.cc.

References Audio3D, fpSrcAudio, clAudio3D::GetWindowSize(), lFragSize, and clAudio3D::Process().

Referenced by OnGdkInput().

7.13.3.13 void clBeamAudioUI::Dither () [private]

Definition at line 560 of file BeamAudioUI.cc.

References BAUI_INT_DATATYPE, DitherBuf, DitherRandBuf, DSP, fpSrcAudio, and lFragSize.

Referenced by OnGdkInput().

7.13.3.14 void clBeamAudioUI::ConvertFromDither () [private]

Definition at line 584 of file BeamAudioUI.cc.

References BAUI_INT_DATATYPE, BAUI_SND_DATATYPE, DitherBuf, ipOutAudio, lBufIdx, and lDataCount.

Referenced by OnGdkInput().

7.13.3.15 int clBeamAudioUI::Main ()

Definition at line 642 of file BeamAudioUI.cc.

Referenced by main().

7.13.3.16 gboolean clBeamAudioUI::OnDelete (GtkWidget *, GdkEvent *, gpointer)

Definition at line 649 of file BeamAudioUI.cc.

References bConnected, and giGdkTag.

Referenced by WrapOnDelete().

7.13.3.17 void clBeamAudioUI::OnConnectClick (GtkButton *, gpointer)

Definition at line 658 of file BeamAudioUI.cc.

References BAUI_SERVER_MAXLEN, bConnected, clSockOp::Close(), giGdkTag, gwCServer, InitConnection(), ParseServerStr(), ptidAudio, and SOp.

Referenced by WrapOnConnectClick().

7.13.3.18 void clBeamAudioUI::OnValueChanged (GtkAdjustment *, gpointer)

Definition at line 690 of file BeamAudioUI.cc.

References DSP, _stBeamAudioReq::fDirection, goADirection, sRequest, and UpdateSettings().

Referenced by WrapOnValueChanged().

7.13.3.19 void clBeamAudioUI::OnToggled (GtkToggleButton *, gpointer)

Definition at line 705 of file BeamAudioUI.cc.

References b3DAudio, bDither, _stBeamAudioReq::bHighFreq, gwCB3DAudio, gwCBDither, gwCBHighFreq, sRequest, and UpdateSettings().

Referenced by WrapOnToggled().

7.13.3.20 void clBeamAudioUI::OnGdkInput (gpointer, gint, GdkInputCondition)

Definition at line 730 of file BeamAudioUI.cc.

References Audio3D, b3DAudio, BAUI_AUDIO_BUFCOUNT, BAUI_CONV_BUF_LEN, bConnected, bDither, ConvertFromDither(), ConvertMS(), Dither(), DSP, _stBeamAudioRes::fDirection, _stBeamAudioRes::fPeakLevel, fpInAudio, fpSrcAudio, clSockOp::GetErrno(), clBeamAudioMsg::GetResult(), guSbCtxt, gwStatusBar, ipOutAudio, iPrevHeading, lBufIdx, lClips, lDataCount, lMsgSize, MessageBuf, Msg, OutBufMutex, Process3D(), clSockOp::ReadN(), clSockOp::ReadSelect(), clAudio3D::SetAngles(), and SOp.

Referenced by WrapOnGdkInput().

7.13.3.21 void * clBeamAudioUI::AudioOutThread (void *)

Definition at line 785 of file BeamAudioUI.cc.

References bALSA, BAUI_AUDIO_BUFCOUNT, BAUI_SND_DATATYPE, bConnected, clAudio::Close(), clAudio::GetOutBufInfo(), InitAudio(), ipOutAudio, lBufIdx, lDataCount, OutBufMutex, and clAudio::Write().

Referenced by WrapAudioOutThread().

7.13.4 Member Data Documentation

7.13.4.1 volatile bool clBeamAudioUI::bConnected [private]

Definition at line 78 of file BeamAudioUI.hh.

Referenced by AudioOutThread(), clBeamAudioUI(), InitConnection(), OnConnectClick(), OnDelete(), OnGdkInput(), and UpdateSettings().

7.13.4.2 bool clBeamAudioUI::bDither [private]

Definition at line 79 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI(), OnGdkInput(), and OnToggled().

7.13.4.3 bool clBeamAudioUI::b3DAudio [private]

Definition at line 80 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI(), OnGdkInput(), and OnToggled().

7.13.4.4 bool clBeamAudioUI::bALSA [private]

Definition at line 81 of file BeamAudioUI.hh.

Referenced by AudioOutThread(), and clBeamAudioUI().

7.13.4.5 int clBeamAudioUI::iSampleRate [private]

Definition at line 82 of file BeamAudioUI.hh.

Referenced by AllocateBuffers().

7.13.4.6 int clBeamAudioUI::iRandH [private]

Definition at line 83 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), clBeamAudioUI(), and ∼clBeamAudioUI().

7.13.4.7 int clBeamAudioUI::iPrevHeading [private]

Definition at line 84 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI(), and OnGdkInput().

7.13.4.8 long clBeamAudioUI::lFragSize [private]

Definition at line 85 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), ConvertMS(), Dither(), and Process3D().

7.13.4.9 long clBeamAudioUI::lDataCount [private]

Definition at line 86 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), AudioOutThread(), ConvertFromDither(), and OnGdkInput().

7.13.4.10 long clBeamAudioUI::lMsgSize [private]

Definition at line 87 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), and OnGdkInput().

7.13.4.11 long clBeamAudioUI::lClips [private]

Definition at line 88 of file BeamAudioUI.hh.

Referenced by InitConnection(), and OnGdkInput().

7.13.4.12 volatile long clBeamAudioUI::lBufIdx [private]

Definition at line 89 of file BeamAudioUI.hh.

Referenced by AudioOutThread(), ConvertFromDither(), OnGdkInput(), and SendSettings().

7.13.4.13 GDT* clBeamAudioUI::fpInAudio [private]

Definition at line 90 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), ConvertMS(), and OnGdkInput().

7.13.4.14 GDT* clBeamAudioUI::fpSrcAudio [private]

Definition at line 91 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), ConvertMS(), Dither(), OnGdkInput(), and Process3D().

7.13.4.15 BAUI_SND_DATATYPE* clBeamAudioUI::ipOutAudio[BAUI_AUDIO_BUFCOUNT] [private]

Definition at line 92 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), AudioOutThread(), ConvertFromDither(), and OnGdkInput().

7.13.4.16 pthread_t clBeamAudioUI::ptidAudio [private]

Definition at line 93 of file BeamAudioUI.hh.

Referenced by OnConnectClick(), and SendSettings().

7.13.4.17 stBeamAudioReq clBeamAudioUI::sRequest [private]

Definition at line 94 of file BeamAudioUI.hh.

Referenced by OnToggled(), OnValueChanged(), SendSettings(), and UpdateSettings().

7.13.4.18 guint clBeamAudioUI::guSbCtxt [private]

Definition at line 96 of file BeamAudioUI.hh.

Referenced by Build(), and OnGdkInput().

7.13.4.19 gint clBeamAudioUI::giGdkTag [private]

Definition at line 97 of file BeamAudioUI.hh.

Referenced by InitConnection(), OnConnectClick(), and OnDelete().

7.13.4.20 GList* clBeamAudioUI::glServer [private]

Definition at line 98 of file BeamAudioUI.hh.

Referenced by BuildTable1().

7.13.4.21 GtkWidget* clBeamAudioUI::gwWindow [private]

Definition at line 100 of file BeamAudioUI.hh.

Referenced by Build(), and ConnectSignals().

7.13.4.22 GtkWidget* clBeamAudioUI::gwVBox [private]

Definition at line 101 of file BeamAudioUI.hh.

Referenced by Build(), BuildTable1(), and BuildTable2().

7.13.4.23 GtkWidget* clBeamAudioUI::gwTable1 [private]

Definition at line 103 of file BeamAudioUI.hh.

Referenced by BuildTable1().

7.13.4.24 GtkWidget* clBeamAudioUI::gwLServer [private]

Definition at line 104 of file BeamAudioUI.hh.

Referenced by BuildTable1().

7.13.4.25 GtkWidget* clBeamAudioUI::gwCServer [private]

Definition at line 105 of file BeamAudioUI.hh.

Referenced by BuildTable1(), and OnConnectClick().

7.13.4.26 GtkWidget* clBeamAudioUI::gwBConnect [private]

Definition at line 106 of file BeamAudioUI.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.13.4.27 GtkWidget* clBeamAudioUI::gwTable2 [private]

Definition at line 108 of file BeamAudioUI.hh.

Referenced by BuildTable2().

7.13.4.28 GtkWidget* clBeamAudioUI::gwLSoundSpeed [private]

Definition at line 109 of file BeamAudioUI.hh.

Referenced by BuildTable2().

7.13.4.29 GtkWidget* clBeamAudioUI::gwESoundSpeed [private]

Definition at line 110 of file BeamAudioUI.hh.

Referenced by BuildTable2(), and SendSettings().

7.13.4.30 GtkWidget* clBeamAudioUI::gwCBHighFreq [private]

Definition at line 111 of file BeamAudioUI.hh.

Referenced by BuildTable2(), ConnectSignals(), OnToggled(), and SendSettings().

7.13.4.31 GtkWidget* clBeamAudioUI::gwCBDither [private]

Definition at line 112 of file BeamAudioUI.hh.

Referenced by BuildTable2(), ConnectSignals(), and OnToggled().

7.13.4.32 GtkWidget* clBeamAudioUI::gwCB3DAudio [private]

Definition at line 113 of file BeamAudioUI.hh.

Referenced by BuildTable2(), ConnectSignals(), and OnToggled().

7.13.4.33 GtkWidget* clBeamAudioUI::gwLDirection [private]

Definition at line 115 of file BeamAudioUI.hh.

Referenced by Build().

7.13.4.34 GtkObject* clBeamAudioUI::goADirection [private]

Definition at line 116 of file BeamAudioUI.hh.

Referenced by Build(), ConnectSignals(), OnValueChanged(), and SendSettings().

7.13.4.35 GtkWidget* clBeamAudioUI::gwHSDirection [private]

Definition at line 117 of file BeamAudioUI.hh.

Referenced by Build().

7.13.4.36 GtkWidget* clBeamAudioUI::gwStatusBar [private]

Definition at line 121 of file BeamAudioUI.hh.

Referenced by Build(), and OnGdkInput().

7.13.4.37 clAlloc clBeamAudioUI::MessageBuf [private]

Definition at line 123 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), and OnGdkInput().

7.13.4.38 clAlloc clBeamAudioUI::InAudioBuf [private]

Definition at line 124 of file BeamAudioUI.hh.

Referenced by AllocateBuffers().

7.13.4.39 clAlloc clBeamAudioUI::SrcAudioBuf [private]

Definition at line 125 of file BeamAudioUI.hh.

Referenced by AllocateBuffers().

7.13.4.40 clAlloc clBeamAudioUI::DitherBuf [private]

Definition at line 126 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), ConvertFromDither(), and Dither().

7.13.4.41 clAlloc clBeamAudioUI::DitherRandBuf [private]

Definition at line 127 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), and Dither().

7.13.4.42 clAlloc clBeamAudioUI::OutAudioBuf[BAUI_AUDIO_-BUFCOUNT] [private]

Definition at line 128 of file BeamAudioUI.hh.

Referenced by AllocateBuffers().

7.13.4.43 clAudio3D clBeamAudioUI::Audio3D [private]

Definition at line 130 of file BeamAudioUI.hh.

Referenced by AllocateBuffers(), OnGdkInput(), and Process3D().

7.13.4.44 clCfgFile clBeamAudioUI::Cfg [private]

Definition at line 131 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI().

7.13.4.45 clDSPOp clBeamAudioUI::DSP [private]

Definition at line 132 of file BeamAudioUI.hh.

Referenced by Dither(), OnGdkInput(), and OnValueChanged().

7.13.4.46 clBeamAudioMsg clBeamAudioUI::Msg [private]

Definition at line 133 of file BeamAudioUI.hh.

Referenced by OnGdkInput(), SendSettings(), and UpdateSettings().

7.13.4.47 clMutex clBeamAudioUI::OutBufMutex[BAUI_AUDIO_-BUFCOUNT] [private]

Definition at line 134 of file BeamAudioUI.hh.

Referenced by AudioOutThread(), and OnGdkInput().

7.13.4.48 clGtkUtils clBeamAudioUI::GtkUtils [private]

Definition at line 135 of file BeamAudioUI.hh.

Referenced by BuildTable1().

7.13.4.49 clSockClie clBeamAudioUI::SClient [private]

Definition at line 136 of file BeamAudioUI.hh.

Referenced by InitConnection().

7.13.4.50 clSockOp clBeamAudioUI::SOp [private]

Definition at line 137 of file BeamAudioUI.hh.

Referenced by InitConnection(), OnConnectClick(), OnGdkInput(), SendSettings(), and UpdateSettings().

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

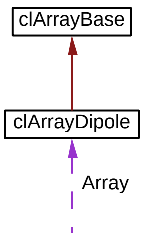
- **BeamAudioUI.hh**
- **BeamAudioUI.cc**

7.14 clBeamDipole Class Reference

Beamformer class for dipole array.

`#include <BeamDipole.hh>`

Collaboration diagram for clBeamDipole:



Public Member Functions

- **clBeamDipole** (**clArrayDipole** *, GDT, int, long, bool)

  Constructor.

- ~**clBeamDipole** ()
- bool **AddData** ()

  Add data to integration buffers.

- GDT **Process** (GDT)

  Process data in integration buffers and return result.

- void **SetHistory** ()

  Set history part of buffer.

- GDT **GetPeakLevel** ()

  Get peak level (in dB) of integration buffer.

- GDT **GetIntegrationTime** ()

  Get real integration time.

Private Attributes

- bool **bDebug**
- long **lSampleCount**

  Sample count for specified integration time.

- long **lBaseIdx**

  Starting index = maximum delay.

- long **lWinSize**

  Window size (input data).

- GDT **fIntTime**

  Integration time in seconds.

- clDSPAlloc **Data** [2]

  Data buffers for integration.

- clDSPOp **DSP**
- clDSPOp **DSPBank** [2]
- **clArrayDipole** * **Array**

7.14.1 Detailed Description

Beamformer class for dipole array.

Definition at line 36 of file BeamDipole.hh.

7.14.2 Constructor & Destructor Documentation

7.14.2.1 clBeamDipole::clBeamDipole (clArrayDipole *, GDT, int, long, bool)

Constructor.

Parameters:

ArrayPtr Pointer to **clArrayDipole**(p. 25)

fIntegrationTime Integration time (s)

iSampleRate Samplerate

lWindowSize Size of window

bEnableDebug Enable debugging output

Definition at line 38 of file BeamDipole.cc.

References Array, bDebug, Data, DSPBank, fIntTime, GDT, clArray-Dipole::GetMaxDelay(), lBaseIdx, lSampleCount, and lWinSize.

7.14.2.2 clBeamDipole::~clBeamDipole ()

Definition at line 68 of file BeamDipole.cc.

7.14.3 Member Function Documentation

7.14.3.1 bool clBeamDipole::AddData ()

Add data to integration buffers.

Call this after calling **clArrayDipole::AddData()**(p. 31) Returns >= 1 when integration buffers are full and ready to be processed by **Process()**(p. 102), call until returns 0 before calling **clArrayDipole::AddData()**(p. 31) again.

Returns:
Status

Definition at line 73 of file BeamDipole.cc.

References Array, Data, DSPBank, GDT, clArrayDipole::GetFiltPtr(), lBaseIdx, lSampleCount, and lWinSize.

Referenced by clDirection::ProcessData().

7.14.3.2 GDT clBeamDipole::Process (GDT)

Process data in integration buffers and return result.

This can be called many times for different directions.

Parameters:
fDirection Direction (rad)

Returns:
Normalized energy

Definition at line 90 of file BeamDipole.cc.

References Array, Data, GDT, clArrayDipole::GetDelaySamples(), lBaseIdx, and lSampleCount.

Referenced by clDirection::CalcDipoleBeams().

7.14.3.3 void clBeamDipole::SetHistory ()

Set history part of buffer.

Call this between last call of **Process()**(p. 102) and next call of **AddData()**(p. 102), this will copy data from end of integration buffer to start of integration buffer.

Definition at line 125 of file BeamDipole.cc.

References Data, DSPBank, GDT, lBaseIdx, and lSampleCount.

Referenced by clDirection::CalcDipoleBeams().

7.14.3.4 GDT clBeamDipole::GetPeakLevel ()

Get peak level (in dB) of integration buffer.

Returns:
Peak level (dB)

Definition at line 139 of file BeamDipole.cc.

References Data, DIR_DB_SCALE, DSPBank, GDT, lBaseIdx, and lSampleCount.

Referenced by clDirection::ProcessData().

7.14.3.5 GDT clBeamDipole::GetIntegrationTime () [inline]

Get real integration time.

Returns:
Integration time (s)

Definition at line 98 of file BeamDipole.hh.

References fIntTime, and GDT.

Referenced by clDirection::InitArray().

7.14.4 Member Data Documentation

7.14.4.1 bool clBeamDipole::bDebug [private]

Definition at line 38 of file BeamDipole.hh.

Referenced by clBeamDipole().

7.14.4.2 long clBeamDipole::lSampleCount [private]

Sample count for specified integration time.

Definition at line 39 of file BeamDipole.hh.

Referenced by AddData(), clBeamDipole(), GetPeakLevel(), Process(), and SetHistory().

7.14.4.3 long clBeamDipole::lBaseIdx [private]

Starting index = maximum delay.

Definition at line 40 of file BeamDipole.hh.

Referenced by AddData(), clBeamDipole(), GetPeakLevel(), Process(), and SetHistory().

7.14.4.4 long clBeamDipole::lWinSize `[private]`

Window size (input data).

Definition at line 41 of file BeamDipole.hh.

Referenced by AddData(), and clBeamDipole().

7.14.4.5 GDT clBeamDipole::fIntTime `[private]`

Integration time in seconds.

Definition at line 42 of file BeamDipole.hh.

Referenced by clBeamDipole(), and GetIntegrationTime().

7.14.4.6 clDSPAlloc clBeamDipole::Data[2] `[private]`

Data buffers for integration.

Definition at line 43 of file BeamDipole.hh.

Referenced by AddData(), clBeamDipole(), GetPeakLevel(), Process(), and SetHistory().

7.14.4.7 clDSPOp clBeamDipole::DSP `[private]`

Definition at line 44 of file BeamDipole.hh.

7.14.4.8 clDSPOp clBeamDipole::DSPBank[2] `[private]`

Definition at line 45 of file BeamDipole.hh.

Referenced by AddData(), clBeamDipole(), GetPeakLevel(), and SetHistory().

7.14.4.9 clArrayDipole* clBeamDipole::Array `[private]`

Definition at line 46 of file BeamDipole.hh.

Referenced by AddData(), clBeamDipole(), and Process().

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

- **BeamDipole.hh**
- **BeamDipole.cc**

7.15 _stBeamNodeParams Struct Reference

Node parameters for beamforming cluster.

`#include <BeamCommon.hh>`

Public Attributes

- int **iType**

  Array type.

- int **iSensors**

  Number of sensors.

- float **fSpacing**

  Sensor spacing.

- float **fSoundSpeed**

  Speed of sound.

- int **iBeamCount**

  Beam count.

- int **iWindowSize**

  Size of window.

- int **iBlockSize**

  Cluster communication block size.

- float **fSampleRate**

  Samplerate of incoming data.

7.15.1 Detailed Description

Node parameters for beamforming cluster.

Definition at line 47 of file BeamCommon.hh.

7.15.2 Member Data Documentation

7.15.2.1 int _stBeamNodeParams::iType

Array type.

Definition at line 49 of file BeamCommon.hh.

Referenced by BeamCommNodeParams(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clBeamSrv2Master::ReadConfig(), and clBeamSrvMaster::ReadConfig().

7.15.2.2 int _stBeamNodeParams::iSensors

Number of sensors.

Definition at line 50 of file BeamCommon.hh.

Referenced by BeamCommNodeParams(), clBeamSrv2Master::InitFilterBank(), clBeamSrvMaster::InitFilterBank(), clBeamSrv2Master::InitProcessing(), clBeamSrvMaster::InitProcessing(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clBeamSrv2Master::ProcessLoop(), clBeamSrvMaster::ProcessLoop(), clBeamSrv2Master::ReadConfig(), and clBeamSrvMaster::ReadConfig().

7.15.2.3 float _stBeamNodeParams::fSpacing

Sensor spacing.

Definition at line 51 of file BeamCommon.hh.

Referenced by BeamCommNodeParams(), clBeamSrv2Master::InitFilterBank(), clBeamSrvMaster::InitFilterBank(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clBeamSrv2Master::ReadConfig(), and clBeamSrvMaster::ReadConfig().

7.15.2.4 float _stBeamNodeParams::fSoundSpeed

Speed of sound.

Definition at line 52 of file BeamCommon.hh.

Referenced by BeamCommNodeParams(), clBeamSrv2Master::InitFilterBank(), clBeamSrvMaster::InitFilterBank(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clBeamSrv2Master::ReadConfig(), and clBeamSrvMaster::ReadConfig().

7.15.2.5 int _stBeamNodeParams::iBeamCount

Beam count.

Definition at line 53 of file BeamCommon.hh.

Referenced by BeamCommNodeParams(), clBeamSrv2Master::InitProcessing(), clBeamSrvMaster::InitProcessing(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clBeamSrv2Master::ProcessLoop(), clBeamSrvMaster::ProcessLoop(), clBeamSrv2Master::ReadConfig(), clBeamSrvMaster::ReadConfig(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), and clBeamSrv2Master::WaitReady().

7.15.2.6 int _stBeamNodeParams::iWindowSize

Size of window.

Definition at line 54 of file BeamCommon.hh.

Referenced by BeamCommNodeParams(), clBeamSrv2Master::InitFilterBank(), clBeamSrvMaster::InitFilterBank(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clBeamSrv2Master::ReadConfig(), and clBeamSrvMaster::ReadConfig().

7.15.2.7 int _stBeamNodeParams::iBlockSize

Cluster communication block size.

Definition at line 55 of file BeamCommon.hh.

Referenced by BeamCommNodeParams(), clBeamSrv2Master::InDataThread(), clBeamSrvMaster::InDataThread(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clBeamSrv2Master::ProcessLoop(), clBeamSrvMaster::ProcessLoop(), clBeamSrv2Master::ReadConfig(), and clBeamSrvMaster::ReadConfig().

7.15.2.8 float _stBeamNodeParams::fSampleRate

Samplerate of incoming data.

Definition at line 56 of file BeamCommon.hh.

Referenced by BeamCommNodeParams(), clBeamSrv2Master::InitFilterBank(), clBeamSrvMaster::InitFilterBank(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clBeamSrv2Master::ServeClientThread(), and clBeamSrvMaster::ServeClientThread().

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

- **BeamCommon.hh**

7.16 _stBeamProcInfo Struct Reference

`#include <BeamSrv.hh>`

Public Attributes

- pid_t **pidProc**
- int **iSockH**

7.16.1 Member Data Documentation

7.16.1.1 pid_t _stBeamProcInfo::pidProc

Definition at line 56 of file BeamSrv.hh.

Referenced by main().

7.16.1.2 int _stBeamProcInfo::iSockH

Definition at line 57 of file BeamSrv.hh.

Referenced by main().

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

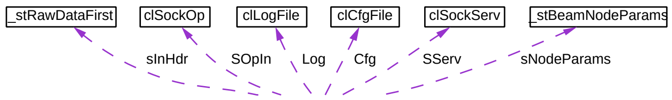
- **BeamSrv.hh**

7.17 clBeamSrv2Master Class Reference

Beamforming input server, root node.

`#include <BeamSrv2.hh>`

Collaboration diagram for clBeamSrv2Master:



Public Member Functions

- **clBeamSrv2Master** ()
- ~**clBeamSrv2Master** ()
- int **Main** (int ∗, char ∗∗∗)
- void ∗ **InDataThread** (void ∗)
- void ∗ **ServerThread** (void ∗)
- void ∗ **ServeClientThread** (void ∗)

Static Public Member Functions

- void **Abort** (int=0)

Private Member Functions

- bool **ReadConfig** ()
- bool **InitFilterBank** ()
- bool **InitProcessing** ()
- bool **WaitReady** ()
- void **ProcessLoop** ()
- void **CompactData** (GDT ∗, const GDT ∗, long, long, long, long)
- bool **FilterData** (GDT ∗, const GDT ∗, long, long, long, long)

Private Attributes

- int **iFilterType**
- int **iDecimate**
- int **iDecFact**
- int **iChOffset**
- int **iOutDataCount**
- char **cpLogBuf** [BS_LOGBUFSIZE]
- **stRawDataFirst sInHdr**

- **stBeamNodeParams sNodeParams**
- clAlloc **FiltWork**
- **clCfgFile Cfg**
- clDSPOp **DSP**
- **clLogFile Log**
- clRecDecimator * **FilterBank**
- **clSockOp SOpIn**
- **clSockServ SServ**
- clReBuffer **InBuf**
- clSemaphore **SemIn**
- clMutex **MtxIn**
- clAlloc **OutData**
- clCondition **CndOut**
- clMutex **MtxOut**

7.17.1 Detailed Description

Beamforming input server, root node.

This node reads data from streamdist, distributes it to slave nodes, collects beamformed data and sends it to clients. Each beam is represented as a channel at output.

Definition at line 63 of file BeamSrv2.hh.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 clBeamSrv2Master::clBeamSrv2Master ()

Definition at line 424 of file BeamSrv2.cc.

References clLogFile::Add(), BS_LOGFILE, FilterBank, Log, and clLogFile::Open().

7.17.2.2 clBeamSrv2Master::~clBeamSrv2Master ()

Definition at line 432 of file BeamSrv2.cc.

References clLogFile::Add(), FilterBank, and Log.

7.17.3 Member Function Documentation

7.17.3.1 bool clBeamSrv2Master::ReadConfig () `[private]`

Definition at line 159 of file BeamSrv2.cc.

References clLogFile::Add(), Cfg, _stBeamNodeParams::fSoundSpeed, _stBeamNodeParams::fSpacing, clCfgFile::GetFlt(), clCfgFile::GetInt(), _stBeamNodeParams::iBeamCount, _stBeamNodeParams::iBlockSize, iDecimate, _stBeamNodeParams::iSensors, _stBeamNodeParams::iType, _stBeamNodeParams::iWindowSize, Log, and sNodeParams.

Referenced by Main().

7.17.3.2 bool clBeamSrv2Master::InitFilterBank () [private]

Definition at line 206 of file BeamSrv2.cc.

References Abort(), clLogFile::Add(), Cfg, cpLogBuf, _stRawDataFirst::dSampleRate, FilterBank, _stBeamNodeParams::fSampleRate, _stBeamNodeParams::fSoundSpeed, _stBeamNodeParams::fSpacing, GDT, clCfgFile::GetInt(), iDecFact, iDecimate, iFilterType, _stBeamNodeParams::iSensors, _stBeamNodeParams::iWindowSize, Log, sInHdr, and sNodeParams.

Referenced by Main().

7.17.3.3 bool clBeamSrv2Master::InitProcessing () [private]

Definition at line 251 of file BeamSrv2.cc.

References clLogFile::Add(), BeamCommNodeParams(), _stBeamNodeParams::iBeamCount, _stRawDataFirst::iChannels, iChOffset, _stBeamNodeParams::iSensors, Log, sInHdr, and sNodeParams.

Referenced by Main().

7.17.3.4 bool clBeamSrv2Master::WaitReady () [private]

Definition at line 272 of file BeamSrv2.cc.

References clLogFile::Add(), BS_TAG_READY, cpLogBuf, _stBeamNodeParams::iBeamCount, Log, and sNodeParams.

Referenced by Main().

7.17.3.5 void clBeamSrv2Master::ProcessLoop () [private]

Definition at line 292 of file BeamSrv2.cc.

References clLogFile::Add(), BeamCommInData(), bRun, BS_TAG_RES, CndOut, CompactData(), cpLogBuf, DSP, FilterData(), GDT, _stBeamNodeParams::iBeamCount, _stBeamNodeParams::iBlockSize, _stRawDataFirst::iChannels, iChOffset, iDecFact, InBuf, iOutDataCount, _stBeamNodeParams::iSensors, Log, MtxIn, MtxOut, OutData, SemIn, sInHdr, and sNodeParams.

Referenced by Main().

7.17.3.6 void clBeamSrv2Master::CompactData (GDT *, const GDT *, long, long, long, long) [private]

Definition at line 374 of file BeamSrv2.cc.

Referenced by ProcessLoop().

7.17.3.7 bool clBeamSrv2Master::FilterData (GDT *, const GDT *, long, long, long, long) [private]

Definition at line 394 of file BeamSrv2.cc.

References DSP, FilterBank, FiltWork, and GDT.

Referenced by ProcessLoop().

7.17.3.8 int clBeamSrv2Master::Main (int *, char ***)

Definition at line 440 of file BeamSrv2.cc.

References Abort(), clLogFile::Add(), BeamSrvMasterThreads, clSockServ::Bind(), BS_CFGFILE, Cfg, clSockClie::Connect(), cpLogBuf, _stRawDataFirst::dSampleRate, clSockOp::GetErrno(), clSockClie::GetErrno(), clSockServ::GetErrno(), clCfgFile::GetInt(), clCfgFile::GetStr(), _stRawDataReq::iChannel, _stRawDataFirst::iChannels, iChOffset, InDataThread(), InitFilterBank(), InitProcessing(), Log, ProcessLoop(), ReadConfig(), clSockOp::ReadN(), ServerThread(), clCfgFile::SetFileName(), clSockOp::SetHandle(), sInHdr, SOpIn, SServ, WaitReady(), and clSockOp::WriteN().

7.17.3.9 void clBeamSrv2Master::Abort (int = 0) [static]

Definition at line 535 of file BeamSrv2.cc.

Referenced by InDataThread(), InitFilterBank(), and Main().

7.17.3.10 void * clBeamSrv2Master::InDataThread (void *)

Definition at line 541 of file BeamSrv2.cc.

References Abort(), bRun, GDT, _stBeamNodeParams::iBlockSize, _stRawDataFirst::iChannels, InBuf, MtxIn, clSockOp::ReadN(), SemIn, sInHdr, sNodeParams, and SOpIn.

Referenced by Main().

7.17.3.11 void * clBeamSrv2Master::ServerThread (void *)

Definition at line 566 of file BeamSrv2.cc.

References BeamSrvMasterThreads, bRun, BS_ACCEPT_TO, ServeClientThread(), SServ, and clSockServ::WaitForConnect().

Referenced by Main().

7.17.3.12 void * clBeamSrv2Master::ServeClientThread (void *)

Definition at line 584 of file BeamSrv2.cc.

References clLogFile::Add(), bRun, CndOut, clSockOp::DisableNagle(), _stSoundStart::dSampleRate, _stBeamNodeParams::fSampleRate, GDT, clSockOp::GetErrno(), GLOBAL_HEADER_LEN, _stBeamNodeParams::iBeamCount, _stSoundStart::iChannels, _stSoundStart::iCompress, _stSoundStart::iFragmentSize, iOutDataCount, Log, MSG_SOUND_COMPRESS_NONE, MtxOut, OutData, clSoundMsg::SetData(), clSockOp::SetHandle(), clSoundMsg::SetStart(), clSockOp::SetTypeOfService(), sNodeParams, SOp, and clSockOp::WriteN().

Referenced by ServerThread().

7.17.4 Member Data Documentation

7.17.4.1 int clBeamSrv2Master::iFilterType [private]

Definition at line 65 of file BeamSrv2.hh.

Referenced by InitFilterBank().

7.17.4.2 int clBeamSrv2Master::iDecimate [private]

Definition at line 66 of file BeamSrv2.hh.

Referenced by InitFilterBank(), and ReadConfig().

7.17.4.3 int clBeamSrv2Master::iDecFact [private]

Definition at line 67 of file BeamSrv2.hh.

Referenced by InitFilterBank(), and ProcessLoop().

7.17.4.4 int clBeamSrv2Master::iChOffset [private]

Definition at line 68 of file BeamSrv2.hh.

Referenced by InitProcessing(), Main(), and ProcessLoop().

7.17.4.5 int clBeamSrv2Master::iOutDataCount [private]

Definition at line 69 of file BeamSrv2.hh.

Referenced by ProcessLoop(), and ServeClientThread().

7.17.4.6 char clBeamSrv2Master::cpLogBuf[BS_LOGBUFSIZE] [private]

Definition at line 70 of file BeamSrv2.hh.

Referenced by InitFilterBank(), Main(), ProcessLoop(), and WaitReady().

7.17.4.7 stRawDataFirst clBeamSrv2Master::sInHdr [private]

Definition at line 71 of file BeamSrv2.hh.

Referenced by InDataThread(), InitFilterBank(), InitProcessing(), Main(), and ProcessLoop().

7.17.4.8 stBeamNodeParams clBeamSrv2Master::sNodeParams [private]

Definition at line 72 of file BeamSrv2.hh.

Referenced by InDataThread(), InitFilterBank(), InitProcessing(), ProcessLoop(), ReadConfig(), ServeClientThread(), and WaitReady().

7.17.4.9 clAlloc clBeamSrv2Master::FiltWork [private]

Definition at line 73 of file BeamSrv2.hh.

Referenced by FilterData().

7.17.4.10 clCfgFile clBeamSrv2Master::Cfg [private]

Definition at line 74 of file BeamSrv2.hh.

Referenced by InitFilterBank(), Main(), and ReadConfig().

7.17.4.11 clDSPOp clBeamSrv2Master::DSP [private]

Definition at line 75 of file BeamSrv2.hh.

Referenced by FilterData(), and ProcessLoop().

7.17.4.12 clLogFile clBeamSrv2Master::Log [private]

Definition at line 76 of file BeamSrv2.hh.

Referenced by clBeamSrv2Master(), InitFilterBank(), InitProcessing(), Main(), ProcessLoop(), ReadConfig(), ServeClientThread(), WaitReady(), and ~clBeamSrv2Master().

7.17.4.13 clRecDecimator∗ clBeamSrv2Master::FilterBank [private]

Definition at line 77 of file BeamSrv2.hh.

Referenced by clBeamSrv2Master(), FilterData(), InitFilterBank(), and ∼clBeamSrv2Master().

7.17.4.14 clSockOp clBeamSrv2Master::SOpIn [private]

Definition at line 78 of file BeamSrv2.hh.

Referenced by InDataThread(), and Main().

7.17.4.15 clSockServ clBeamSrv2Master::SServ [private]

Definition at line 79 of file BeamSrv2.hh.

Referenced by Main(), and ServerThread().

7.17.4.16 clReBuffer clBeamSrv2Master::InBuf [private]

Definition at line 81 of file BeamSrv2.hh.

Referenced by InDataThread(), and ProcessLoop().

7.17.4.17 clSemaphore clBeamSrv2Master::SemIn [private]

Definition at line 82 of file BeamSrv2.hh.

Referenced by InDataThread(), and ProcessLoop().

7.17.4.18 clMutex clBeamSrv2Master::MtxIn [private]

Definition at line 83 of file BeamSrv2.hh.

Referenced by InDataThread(), and ProcessLoop().

7.17.4.19 clAlloc clBeamSrv2Master::OutData [private]

Definition at line 85 of file BeamSrv2.hh.

Referenced by ProcessLoop(), and ServeClientThread().

7.17.4.20 clCondition clBeamSrv2Master::CndOut [private]

Definition at line 86 of file BeamSrv2.hh.

Referenced by ProcessLoop(), and ServeClientThread().

7.17.4.21 clMutex clBeamSrv2Master::MtxOut [private]

Definition at line 87 of file BeamSrv2.hh.

Referenced by ProcessLoop(), and ServeClientThread().

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

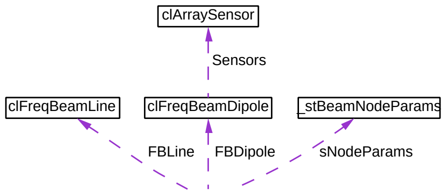
- **BeamSrv2.hh**
- **BeamSrv2.cc**

7.18 clBeamSrv2Slave Class Reference

Beamforming input server, slave node.

`#include <BeamSrv2.hh>`

Collaboration diagram for clBeamSrv2Slave:



Public Member Functions

- **clBeamSrv2Slave** (int)
- ~**clBeamSrv2Slave** ()
- int **Main** (int ∗, char ∗∗∗)

Private Member Functions

- bool **SendReady** ()
- bool **SendRes** (GDT ∗, int)

Private Attributes

- int **iRank**
- **stBeamNodeParams sNodeParams**
- **clFreqBeamDipole FBDipole**
- **clFreqBeamLine FBLine**

7.18.1 Detailed Description

Beamforming input server, slave node.

Slave nodes do the actual beam processing.

Definition at line 112 of file BeamSrv2.hh.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 clBeamSrv2Slave::clBeamSrv2Slave (int)

Definition at line 667 of file BeamSrv2.cc.

References iRank.

7.18.2.2 clBeamSrv2Slave::~clBeamSrv2Slave ()

Definition at line 673 of file BeamSrv2.cc.

7.18.3 Member Function Documentation

7.18.3.1 bool clBeamSrv2Slave::SendReady () [private]

Definition at line 640 of file BeamSrv2.cc.

References BS_TAG_READY, and iRank.

Referenced by Main().

7.18.3.2 bool clBeamSrv2Slave::SendRes (GDT *, int) [private]

Definition at line 649 of file BeamSrv2.cc.

References BS_TAG_RES, and GDT.

Referenced by Main().

7.18.3.3 int clBeamSrv2Slave::Main (int *, char ***)

Definition at line 678 of file BeamSrv2.cc.

References BeamCommInData(), BeamCommNodeParams(), bRun, BS_ARRAY_TYPE_CYLINDER, BS_ARRAY_TYPE_DIPOLE, BS_ARRAY_TYPE_LINE, BS_ARRAY_TYPE_PLANE, BS_ARRAY_TYPE_SPHERE, BS_ARRAY_TYPE_TRIANGLE, FBDipole, FBLine, _stBeamNodeParams::fSampleRate, _stBeamNodeParams::fSoundSpeed, _stBeamNodeParams::fSpacing, GDT, clFreqBeamLine::Get(), clFreqBeamDipole::Get(), _stBeamNodeParams::iBeamCount, _stBeamNodeParams::iBlockSize, clFreqBeamLine::Initialize(), clFreqBeamDipole::Initialize(), iRank, _stBeamNodeParams::iSensors, _stBeamNodeParams::iType, _stBeamNodeParams::iWindowSize, clFreqBeamLine::Put(), clFreqBeamDipole::Put(), SendReady(), SendRes(), clFreqBeamLine::SetDirection(), clFreqBeamDipole::SetDirection(), clFreqBeamLine::SetSoundSpeed(), clFreqBeamDipole::SetSoundSpeed(), and sNodeParams.

Referenced by main().

7.18.4 Member Data Documentation

7.18.4.1 int clBeamSrv2Slave::iRank [private]

Definition at line 114 of file BeamSrv2.hh.

Referenced by clBeamSrv2Slave(), Main(), and SendReady().

7.18.4.2 stBeamNodeParams clBeamSrv2Slave::sNodeParams [private]

Definition at line 115 of file BeamSrv2.hh.

Referenced by Main().

7.18.4.3 clFreqBeamDipole clBeamSrv2Slave::FBDipole [private]

Definition at line 116 of file BeamSrv2.hh.

Referenced by Main().

7.18.4.4 clFreqBeamLine clBeamSrv2Slave::FBLine [private]

Definition at line 117 of file BeamSrv2.hh.

Referenced by Main().

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

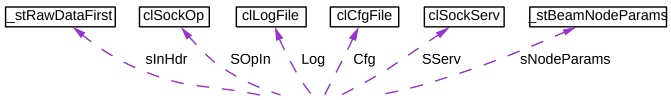
- **BeamSrv2.hh**
- **BeamSrv2.cc**

7.19 clBeamSrvMaster Class Reference

Beamforming input server, main process.

`#include <BeamSrv.hh>`

Collaboration diagram for clBeamSrvMaster:



Public Member Functions

- **clBeamSrvMaster** ()
- ~**clBeamSrvMaster** ()
- int **Main** (int *, char ***)
- void * **InDataThread** (void *)
- void * **ServerThread** (void *)
- void * **ServeClientThread** (void *)

Static Public Member Functions

- void **Abort** ()

Private Member Functions

- bool **ReadConfig** ()
- bool **InitFilterBank** ()
- bool **InitProcessing** ()
- void **ProcessLoop** ()
- bool **SendNodeParams** ()
- bool **WaitReady** ()
- bool **SendInData** (const GDT *, int)
- void **CompactData** (GDT *, const GDT *, long, long, long, long)
- bool **FilterData** (GDT *, const GDT *, long, long, long, long)

Private Attributes

- int **iFilterType**
- int **iDecimate**
- int **iDecFact**
- int **iChOffset**
- int **iOutDataCount**

- volatile int **iBlockCntr**
- char **cpLogBuf** [BS_LOGBUFSIZE]
- **stRawDataFirst sInHdr**
- **stBeamNodeParams sNodeParams**
- clAlloc **FiltWork**
- **clCfgFile Cfg**
- clDSPOp **DSP**
- **clLogFile Log**
- clRecDecimator * **FilterBank**
- **clSockOp SOpIn**
- **clSockServ SServ**
- clReBuffer **InBuf**
- clSemaphore **SemIn**
- clMutex **MtxIn**
- clAlloc **OutData**
- clCondition **CndOut**
- clMutex **MtxOut**

7.19.1 Detailed Description

Beamforming input server, main process.

This process reads data from streamdist, distributes it to slave processes, collects beamformed data and sends it to clients. Each beam is represented as a channel at output.

Definition at line 68 of file BeamSrv.hh.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 clBeamSrvMaster::clBeamSrvMaster ()

Definition at line 501 of file BeamSrv.cc.

References clLogFile::Add(), BS_LOGFILE, FilterBank, iBlockCntr, Log, and clLogFile::Open().

7.19.2.2 clBeamSrvMaster::~clBeamSrvMaster ()

Definition at line 510 of file BeamSrv.cc.

References clLogFile::Add(), FilterBank, and Log.

7.19.3 Member Function Documentation

7.19.3.1 bool clBeamSrvMaster::ReadConfig () `[private]`

Definition at line 168 of file BeamSrv.cc.

References clLogFile::Add(), Cfg, _stBeamNodeParams::fSoundSpeed, _stBeamNodeParams::fSpacing, clCfgFile::GetFlt(), clCfgFile::GetInt(), _stBeamNodeParams::iBeamCount, _stBeamNodeParams::iBlockSize, iDecimate, _stBeamNodeParams::iSensors, _stBeamNodeParams::iType, _stBeamNodeParams::iWindowSize, Log, and sNodeParams.

Referenced by Main().

7.19.3.2 bool clBeamSrvMaster::InitFilterBank () [private]

Definition at line 215 of file BeamSrv.cc.

References Abort(), clLogFile::Add(), Cfg, cpLogBuf, _stRawDataFirst::dSampleRate, FilterBank, _stBeamNodeParams::fSampleRate, _stBeamNodeParams::fSoundSpeed, _stBeamNodeParams::fSpacing, GDT, clCfgFile::GetInt(), iDecFact, iDecimate, iFilterType, _stBeamNodeParams::iSensors, _stBeamNodeParams::iWindowSize, Log, sInHdr, and sNodeParams.

Referenced by Main().

7.19.3.3 bool clBeamSrvMaster::InitProcessing () [private]

Definition at line 262 of file BeamSrv.cc.

References clLogFile::Add(), _stBeamNodeParams::iBeamCount, _stRawDataFirst::iChannels, iChOffset, _stBeamNodeParams::iSensors, Log, SendNodeParams(), sInHdr, sNodeParams, and vProcInfo.

Referenced by Main().

7.19.3.4 void clBeamSrvMaster::ProcessLoop () [private]

Definition at line 280 of file BeamSrv.cc.

References clLogFile::Add(), bRun, CndOut, CompactData(), cpLogBuf, DSP, FilterData(), GDT, _stBeamNodeParams::iBeamCount, iBlockCntr, _stBeamNodeParams::iBlockSize, _stRawDataFirst::iChannels, iChOffset, iDecFact, InBuf, iOutDataCount, _stBeamNodeParams::iSensors, Log, MtxIn, MtxOut, OutData, SemIn, SendInData(), sInHdr, sNodeParams, and vProcInfo.

Referenced by Main().

7.19.3.5 bool clBeamSrvMaster::SendNodeParams () [private]

Definition at line 363 of file BeamSrv.cc.

References clLogFile::Add(), cpLogBuf, Log, sNodeParams, and vProcInfo.

Referenced by InitProcessing().

7.19.3.6 bool clBeamSrvMaster::WaitReady () [private]

Definition at line 391 of file BeamSrv.cc.

References clLogFile::Add(), cpLogBuf, Log, and vProcInfo.

Referenced by Main().

7.19.3.7 bool clBeamSrvMaster::SendInData (const GDT *, int) [private]

Definition at line 422 of file BeamSrv.cc.

References clLogFile::Add(), cpLogBuf, GDT, Log, and vProcInfo.

Referenced by ProcessLoop().

7.19.3.8 void clBeamSrvMaster::CompactData (GDT *, const GDT *, long, long, long, long) [private]

Definition at line 449 of file BeamSrv.cc.

Referenced by ProcessLoop().

7.19.3.9 bool clBeamSrvMaster::FilterData (GDT *, const GDT *, long, long, long, long) [private]

Definition at line 469 of file BeamSrv.cc.

References DSP, FilterBank, FiltWork, and GDT.

Referenced by ProcessLoop().

7.19.3.10 int clBeamSrvMaster::Main (int *, char ***)

Definition at line 518 of file BeamSrv.cc.

References Abort(), clLogFile::Add(), BeamSrvMasterThreads, clSockServ::Bind(), BS_CFGFILE, Cfg, clSockClie::Connect(), cpLogBuf, _stRawDataFirst::dSampleRate, clSockOp::GetErrno(), clSockClie::GetErrno(), clSockServ::GetErrno(), clCfgFile::GetInt(), clCfgFile::GetStr(), _stRawDataReq::iChannel, _stRawDataFirst::iChannels, iChOffset, InDataThread(), InitFilterBank(), InitProcessing(), Log, ProcessLoop(), ReadConfig(), clSockOp::ReadN(), ServerThread(), clCfgFile::SetFileName(), clSockOp::SetHandle(), sInHdr, SOpIn, SServ, WaitReady(), and clSockOp::WriteN().

7.19.3.11 void clBeamSrvMaster::Abort () [static]

Definition at line 613 of file BeamSrv.cc.

References bRun.

Referenced by InDataThread(), InitFilterBank(), and Main().

7.19.3.12 void * clBeamSrvMaster::InDataThread (void *)

Definition at line 619 of file BeamSrv.cc.

References Abort(), BeamSrvMasterThreads, bRun, BS_INTHREAD_-PRIORITY, GDT, _stBeamNodeParams::iBlockSize, _stRawDataFirst::i-Channels, InBuf, MtxIn, clSockOp::ReadN(), SemIn, sInHdr, sNodeParams, and SOpIn.

Referenced by Main().

7.19.3.13 void * clBeamSrvMaster::ServerThread (void *)

Definition at line 647 of file BeamSrv.cc.

References BeamSrvMasterThreads, bRun, BS_ACCEPT_TO, ServeClient-Thread(), SServ, and clSockServ::WaitForConnect().

Referenced by Main().

7.19.3.14 void * clBeamSrvMaster::ServeClientThread (void *)

Definition at line 665 of file BeamSrv.cc.

References clLogFile::Add(), BeamSrvMasterThreads, bRun, BS_-OUTTHREAD_PRIORITY, CndOut, clSockOp::DisableNagle(), _stSound-Start::dSampleRate, _stBeamNodeParams::fSampleRate, GDT, clSockOp::Get-Errno(), GLOBAL_HEADER_LEN, _stBeamNodeParams::iBeamCount, iBlockCntr, _stSoundStart::iChannels, _stSoundStart::iCompress, _stSound-Start::iFragmentSize, iOutDataCount, Log, MSG_SOUND_COMPRESS_-NONE, MtxOut, OutData, clSoundMsg::SetData(), clSockOp::SetHandle(), clSoundMsg::SetStart(), clSockOp::SetTypeOfService(), sNodeParams, SOp, and clSockOp::WriteN().

Referenced by ServerThread().

7.19.4 Member Data Documentation

7.19.4.1 int clBeamSrvMaster::iFilterType [private]

Definition at line 70 of file BeamSrv.hh.

Referenced by InitFilterBank().

7.19.4.2 int clBeamSrvMaster::iDecimate [private]

Definition at line 71 of file BeamSrv.hh.

Referenced by InitFilterBank(), and ReadConfig().

7.19.4.3 int clBeamSrvMaster::iDecFact [private]

Definition at line 72 of file BeamSrv.hh.

Referenced by InitFilterBank(), and ProcessLoop().

7.19.4.4 int clBeamSrvMaster::iChOffset [private]

Definition at line 73 of file BeamSrv.hh.

Referenced by InitProcessing(), Main(), and ProcessLoop().

7.19.4.5 int clBeamSrvMaster::iOutDataCount [private]

Definition at line 74 of file BeamSrv.hh.

Referenced by ProcessLoop(), and ServeClientThread().

7.19.4.6 volatile int clBeamSrvMaster::iBlockCntr [private]

Definition at line 75 of file BeamSrv.hh.

Referenced by clBeamSrvMaster(), ProcessLoop(), and ServeClientThread().

7.19.4.7 char clBeamSrvMaster::cpLogBuf[BS_LOGBUFSIZE] [private]

Definition at line 76 of file BeamSrv.hh.

Referenced by InitFilterBank(), Main(), ProcessLoop(), SendInData(), SendNodeParams(), and WaitReady().

7.19.4.8 stRawDataFirst clBeamSrvMaster::sInHdr [private]

Definition at line 77 of file BeamSrv.hh.

Referenced by InDataThread(), InitFilterBank(), InitProcessing(), Main(), and ProcessLoop().

7.19.4.9 stBeamNodeParams clBeamSrvMaster::sNodeParams [private]

Definition at line 78 of file BeamSrv.hh.

Referenced by InDataThread(), InitFilterBank(), InitProcessing(), ProcessLoop(), ReadConfig(), SendNodeParams(), and ServeClientThread().

7.19.4.10 clAlloc clBeamSrvMaster::FiltWork [private]

Definition at line 79 of file BeamSrv.hh.

Referenced by FilterData().

7.19.4.11 clCfgFile clBeamSrvMaster::Cfg [private]

Definition at line 80 of file BeamSrv.hh.

Referenced by InitFilterBank(), Main(), and ReadConfig().

7.19.4.12 clDSPOp clBeamSrvMaster::DSP [private]

Definition at line 81 of file BeamSrv.hh.

Referenced by FilterData(), and ProcessLoop().

7.19.4.13 clLogFile clBeamSrvMaster::Log [private]

Definition at line 82 of file BeamSrv.hh.

Referenced by clBeamSrvMaster(), InitFilterBank(), InitProcessing(), Main(), ProcessLoop(), ReadConfig(), SendInData(), SendNodeParams(), ServeClientThread(), WaitReady(), and ~clBeamSrvMaster().

7.19.4.14 clRecDecimator* clBeamSrvMaster::FilterBank [private]

Definition at line 83 of file BeamSrv.hh.

Referenced by clBeamSrvMaster(), FilterData(), InitFilterBank(), and ~clBeamSrvMaster().

7.19.4.15 clSockOp clBeamSrvMaster::SOpIn [private]

Definition at line 84 of file BeamSrv.hh.

Referenced by InDataThread(), and Main().

7.19.4.16 clSockServ clBeamSrvMaster::SServ [private]

Definition at line 85 of file BeamSrv.hh.

Referenced by Main(), and ServerThread().

7.19.4.17 clReBuffer clBeamSrvMaster::InBuf [private]

Definition at line 87 of file BeamSrv.hh.

Referenced by InDataThread(), and ProcessLoop().

7.19.4.18 clSemaphore clBeamSrvMaster::SemIn [private]

Definition at line 88 of file BeamSrv.hh.

Referenced by InDataThread(), and ProcessLoop().

7.19.4.19 clMutex clBeamSrvMaster::MtxIn [private]

Definition at line 89 of file BeamSrv.hh.

Referenced by InDataThread(), and ProcessLoop().

7.19.4.20 clAlloc clBeamSrvMaster::OutData [private]

Definition at line 91 of file BeamSrv.hh.

Referenced by ProcessLoop(), and ServeClientThread().

7.19.4.21 clCondition clBeamSrvMaster::CndOut [private]

Definition at line 92 of file BeamSrv.hh.

Referenced by ProcessLoop(), and ServeClientThread().

7.19.4.22 clMutex clBeamSrvMaster::MtxOut [private]

Definition at line 93 of file BeamSrv.hh.

Referenced by ProcessLoop(), and ServeClientThread().

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

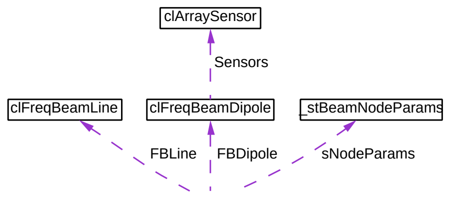
- **BeamSrv.hh**
- **BeamSrv.cc**

7.20 clBeamSrvSlave Class Reference

Beamforming input server, slave process.

`#include <BeamSrv.hh>`

Collaboration diagram for clBeamSrvSlave:



Public Member Functions

- **clBeamSrvSlave** (int, int)
- ∼**clBeamSrvSlave** ()
- int **Main** (int ∗, char ∗∗∗)

Private Member Functions

- bool **RecvParams** ()
- bool **RecvInData** (GDT ∗, int)
- bool **SendReady** ()
- bool **SendRes** (const GDT ∗, int)

Private Attributes

- int **iProcess**
- int **iSockH**
- **stBeamNodeParams sNodeParams**
- **clFreqBeamDipole FBDipole**
- **clFreqBeamLine FBLine**

7.20.1 Detailed Description

Beamforming input server, slave process.

Slave processes do the actual beam processing.

Definition at line 120 of file BeamSrv.hh.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 clBeamSrvSlave::clBeamSrvSlave (int, int)

Definition at line 792 of file BeamSrv.cc.

References iProcess, and iSockH.

7.20.2.2 clBeamSrvSlave::~clBeamSrvSlave ()

Definition at line 799 of file BeamSrv.cc.

References iSockH.

7.20.3 Member Function Documentation

7.20.3.1 bool clBeamSrvSlave::RecvParams () `[private]`

Definition at line 736 of file BeamSrv.cc.

References iSockH, and sNodeParams.

Referenced by Main().

7.20.3.2 bool clBeamSrvSlave::RecvInData (GDT *, int) `[private]`

Definition at line 750 of file BeamSrv.cc.

References GDT, and iSockH.

Referenced by Main().

7.20.3.3 bool clBeamSrvSlave::SendReady () `[private]`

Definition at line 765 of file BeamSrv.cc.

References iProcess, and iSockH.

Referenced by Main().

7.20.3.4 bool clBeamSrvSlave::SendRes (const GDT *, int) `[private]`

Definition at line 777 of file BeamSrv.cc.

References GDT, and iSockH.

Referenced by Main().

7.20.3.5 int clBeamSrvSlave::Main (int ∗, char ∗∗∗)

Definition at line 805 of file BeamSrv.cc.

References bRun, BS_ARRAY_TYPE_CYLINDER, BS_ARRAY_TYPE_DIPOLE, BS_ARRAY_TYPE_LINE, BS_ARRAY_TYPE_PLANE, BS_ARRAY_TYPE_SPHERE, BS_ARRAY_TYPE_TRIANGLE, FBDipole, FBLine, _stBeamNodeParams::fSampleRate, _stBeamNodeParams::fSoundSpeed, _stBeamNodeParams::fSpacing, GDT, clFreqBeamLine::Get(), clFreqBeamDipole::Get(), _stBeamNodeParams::iBeamCount, _stBeamNodeParams::iBlockSize, clFreqBeamLine::Initialize(), clFreqBeamDipole::Initialize(), iProcess, _stBeamNodeParams::iSensors, _stBeamNodeParams::iType, _stBeamNodeParams::iWindowSize, clFreqBeamLine::Put(), clFreqBeamDipole::Put(), RecvInData(), RecvParams(), SendReady(), SendRes(), clFreqBeamLine::SetDirection(), clFreqBeamDipole::SetDirection(), clFreqBeamLine::SetSoundSpeed(), clFreqBeamDipole::SetSoundSpeed(), and sNodeParams.

Referenced by main().

7.20.4 Member Data Documentation

7.20.4.1 int clBeamSrvSlave::iProcess [private]

Definition at line 122 of file BeamSrv.hh.

Referenced by clBeamSrvSlave(), Main(), and SendReady().

7.20.4.2 int clBeamSrvSlave::iSockH [private]

Definition at line 123 of file BeamSrv.hh.

Referenced by clBeamSrvSlave(), RecvInData(), RecvParams(), SendReady(), SendRes(), and ∼clBeamSrvSlave().

7.20.4.3 stBeamNodeParams clBeamSrvSlave::sNodeParams [private]

Definition at line 124 of file BeamSrv.hh.

Referenced by Main(), and RecvParams().

7.20.4.4 clFreqBeamDipole clBeamSrvSlave::FBDipole [private]

Definition at line 125 of file BeamSrv.hh.

Referenced by Main().

7.20.4.5 clFreqBeamLine clBeamSrvSlave::FBLine [private]

Definition at line 126 of file BeamSrv.hh.

Referenced by Main().

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

- **BeamSrv.hh**
- **BeamSrv.cc**

7.21 clCfgFile Class Reference

`#include <CfgFile.hh>`

Public Member Functions

- **clCfgFile** ()
- **clCfgFile** (const char ∗)
- ∼**clCfgFile** ()
- void **SetFileName** (const char ∗)

    Read configuration file.

- bool **GetStr** (const char ∗, char ∗)

    Get value for key.

- bool **GetInt** (const char ∗, int ∗)
- bool **GetInt** (const char ∗, long ∗)
- bool **GetFlt** (const char ∗, float ∗)
- bool **GetFlt** (const char ∗, double ∗)
- int **GetFltArray** (const char ∗, float ∗)
- int **GetFltArray** (const char ∗, double ∗)

Private Member Functions

- void **ReadFile** (const char ∗)
- void **FreeAll** ()

Private Attributes

- int **iEntryCount**
- clAlloc **Names** [CFGF_MAX_ENTRIES]
- clAlloc **Values** [CFGF_MAX_ENTRIES]

7.21.1 Constructor & Destructor Documentation

7.21.1.1 clCfgFile::clCfgFile ()

Definition at line 83 of file CfgFile.cc.

References iEntryCount.

7.21.1.2 clCfgFile::clCfgFile (const char ∗)

Definition at line 89 of file CfgFile.cc.

References iEntryCount, and ReadFile().

7.21.1.3 clCfgFile::~clCfgFile ()

Definition at line 103 of file CfgFile.cc.

References FreeAll().

7.21.2 Member Function Documentation

7.21.2.1 void clCfgFile::ReadFile (const char *) [private]

Definition at line 30 of file CfgFile.cc.

References CFGF_BUFSIZE, CFGF_MAX_ENTRIES, iEntryCount, Names, and Values.

Referenced by clCfgFile(), and SetFileName().

7.21.2.2 void clCfgFile::FreeAll () [private]

Definition at line 70 of file CfgFile.cc.

References iEntryCount, Names, and Values.

Referenced by SetFileName(), and ~clCfgFile().

7.21.2.3 void clCfgFile::SetFileName (const char *)

Read configuration file.

You can call this many times. Previous settings are flushed from the memory every time this method is called.

Parameters:
cpFileName Name of configuration file

Definition at line 96 of file CfgFile.cc.

References FreeAll(), and ReadFile().

Referenced by clBeamAudioUI::clBeamAudioUI(), clComediSrv::clComediSrv(), clGUILofar::clGUILofar(), clSoundProxy::clSoundProxy(), clSoundSrv2::clSoundSrv2(), clSoundSrvA::clSoundSrvA(), clStreamDist::clStreamDist(), clSpectrum::GetCfg(), clSoundUI::GetCfg(), clLevel::GetCfg(), clGUITransient::GetCfg(), clGUILocate::GetCfg(), clGUILevel::GetCfg(), clDirection3::GetCfg(), clDirection2::GetCfg(), clBeamAudio::GetCfg(), clLofarDemon::GetDataHdr(), clSaveSrv::Initialize(), clLofarDemon::Initialize(), clLocate::Initialize(), clFileSrv::Main(), clBeamSrv2Master::Main(), and clBeamSrvMaster::Main().

7.21.2.4 bool clCfgFile::GetStr (const char *, char *)

Get value for key.

Parameters:

cpKey Key name

cpVal Value

Returns:

Success

Definition at line 115 of file CfgFile.cc.

References iEntryCount, Names, and Values.

Referenced by clStreamDist::AudioInThread(), clSoundProxy::clSoundProxy(), clDirection::Exec(), clSoundSrv2::GetAudioCfg(), clComediSrv::GetAudioCfg(), clSpectrum::GetCfg(), clSoundUI::GetCfg(), clLevel::GetCfg(), clDirection3::GetCfg(), clDirection2::GetCfg(), clBeamAudio::GetCfg(), clLofarDemon::GetDataHdr(), clSaveSrv::Initialize(), clStreamDist::Main(), clBeamSrv2Master::Main(), and clBeamSrvMaster::Main().

7.21.2.5 bool clCfgFile::GetInt (const char *, int *)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 131 of file CfgFile.cc.

References iEntryCount, Names, and Values.

Referenced by clStreamDist::AudioInThread(), clSoundSrvA::AudioInThread(), clSoundSrv2::AudioInThread(), clComediSrv::AudioInThread(), clSpectGUI::Build(), clSpectGUI::BuildDrawingPrims(), clGUILofar::BuildDrawingPrims(), clBeamAudioUI::clBeamAudioUI(), clGUIDir::clGUIDir(), clGUILofar::clGUILofar(), clSoundProxy::clSoundProxy(), clSoundSrv::clSoundSrv(), clSpectGUI::clSpectGUI(), clUIServer::clUIServer(), clSaveSrv::CreateFile2(), clGUIDir::Exec(), clDirection::Exec(), clSoundSrvA::GetAudioCfg(), clSoundSrv2::GetAudioCfg(), clComediSrv::GetAudioCfg(), clSpectrum::GetCfg(), clSoundUI::GetCfg(), clLevel::GetCfg(), clGUILocate::GetCfg(), clGUILevel::GetCfg(), clDirection3::GetCfg(), clDirection2::GetCfg(), clBeamAudio::GetCfg(), clDirection::InitArray(), clSoundSrvA::InitCompress(), clSoundSrv2::InitCompress(), clComediSrv::InitCompress(), clLofarDemon::InitFFT(), clLofarDemon::InitFilter(), clBeamSrv2Master::InitFilterBank(), clBeamSrvMaster::InitFilterBank(), clSaveSrv::Initialize(), clLocate::Initialize(), clSoundSrvA::Main(), clSoundSrv2::Main(), clFileSrv::Main(), clComediSrv::Main(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), clBeamSrv2Master::ReadConfig(), clBeamSrvMaster::ReadConfig(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clComediSrv::ServeClientThread(), and clSoundSrv::WaitConnectExec().

7.21.2.6 bool clCfgFile::GetInt (const char *, long *)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 147 of file CfgFile.cc.

References iEntryCount, Names, and Values.

7.21.2.7 bool clCfgFile::GetFlt (const char *, float *)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 163 of file CfgFile.cc.

References iEntryCount, Names, and Values.

Referenced by clGUILofar::clGUILofar(), clGUIDir::Exec(), clComediSrv::GetAudioCfg(), clGUILevel::GetCfg(), clDirection3::GetCfg(), clDirection2::GetCfg(), clBeamAudio::GetCfg(), clDirection::InitArray(), clLocate::Initialize(), clBeamSrv2Master::ReadConfig(), and clBeamSrvMaster::ReadConfig().

7.21.2.8 bool clCfgFile::GetFlt (const char *, double *)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 179 of file CfgFile.cc.

References iEntryCount, Names, and Values.

7.21.2.9 int clCfgFile::GetFltArray (const char *, float *)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 195 of file CfgFile.cc.

References iEntryCount, Names, and Values.

Referenced by clDirection::InitArray().

7.21.2.10 int clCfgFile::GetFltArray (const char *, double *)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

7.21.3 Member Data Documentation

7.21.3.1 int clCfgFile::iEntryCount [private]

Definition at line 36 of file CfgFile.hh.

Referenced by clCfgFile(), FreeAll(), GetFlt(), GetFltArray(), GetInt(), GetStr(), and ReadFile().

7.21.3.2 clAlloc clCfgFile::Names[CFGF_MAX_ENTRIES] [private]

Definition at line 37 of file CfgFile.hh.

Referenced by FreeAll(), GetFlt(), GetFltArray(), GetInt(), GetStr(), and ReadFile().

7.21.3.3 clAlloc clCfgFile::Values[CFGF_MAX_ENTRIES] [private]

Definition at line 38 of file CfgFile.hh.

Referenced by FreeAll(), GetFlt(), GetFltArray(), GetInt(), GetStr(), and ReadFile().

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

- **CfgFile.hh**
- **CfgFile.cc**

7.22 clComediIO Class Reference

Class for Comedi IO operations.

`#include <ComediIO.hh>`

Public Types

- enum { **VERS_MASK_MAJ** = 0x00ff0000, **VERS_MASK_MIN** = 0x0000ff00, **VERS_MASK_PL** = 0x000000ff }
- enum { **VERS_SHIFT_MAJ** = 16, **VERS_SHIFT_MIN** = 8, **VERS_SHIFT_PL** = 0 }

Public Member Functions

- **clComediIO** ()
- ~**clComediIO** ()
- bool **Open** (const char *)

  Open device.

- bool **Close** ()

  Close device.

- int **GetSubdeviceCount** ()

  Get number of subdevices.

- int **GetVersionCode** ()

  Get version code.

- const char * **GetDriverName** ()

  Get driver name.

- const char * **GetBoardName** ()

  Get board name.

- const char * **GetErrorMsg** ()

  Get error message.

- bool **PcmInOpen** ()

  Open PCM input device.

- bool **PcmInClose** ()

  Close PCM input device.

- int **PcmInGetChannelCount** ()

  Get number of channels for PCM input device.

- bool **PcmInStart** (double *, int, double, bool=true, bool=false)

  Start input operation.

- bool **PcmInStop** ()

  Stop input operation.

- long **PcmInRead** (float *, long)

  Read input samples.

- long **PcmInRead** (double *, long)
- bool **PcmBufferSizeSet** (unsigned int)

  Set buffer size for PCM device.

- int **PcmBufferSizeGet** ()

  Get buffer size for PCM device.

- size_t **PcmGetIntSampleSize** ()

  Get internal sample size (for calculating buffer sizes).

Private Member Functions

- void **SetError** ()
- bool **PcmStart** (int, double *, int, unsigned int *, unsigned int, bool, bool)
- long **PcmRead** (sampl_t **, long)

Private Attributes

- int **iInSubDev**
- char * **cpErrorMsg**
- comedi_t * **comediDev**
- lsampl_t **iInMaxValue**
- clAlloc **InScanList**
- clAlloc **InBuf**

7.22.1 Detailed Description

Class for Comedi IO operations.

Definition at line 36 of file ComediIO.hh.

7.22.2 Member Enumeration Documentation

7.22.2.1 anonymous enum

Enumeration values:

VERS_MASK_MAJ

VERS_MASK_MIN

VERS_MASK_PL

Definition at line 49 of file ComediIO.hh.

7.22.2.2 anonymous enum

Enumeration values:

VERS_SHIFT_MAJ

VERS_SHIFT_MIN

VERS_SHIFT_PL

Definition at line 55 of file ComediIO.hh.

7.22.3 Constructor & Destructor Documentation

7.22.3.1 clComediIO::clComediIO ()

Definition at line 108 of file ComediIO.cc.

References comediDev, cpErrorMsg, and iInSubDev.

7.22.3.2 clComediIO::~clComediIO ()

Definition at line 116 of file ComediIO.cc.

References Close().

7.22.4 Member Function Documentation

7.22.4.1 void clComediIO::SetError () `[private]`

Definition at line 30 of file ComediIO.cc.

References cpErrorMsg.

Referenced by Close(), GetVersionCode(), Open(), PcmBufferSizeGet(), PcmBufferSizeSet(), PcmInClose(), PcmInGetChannelCount(), PcmInOpen(), PcmInRead(), PcmInStart(), PcmInStop(), and PcmStart().

7.22.4.2 bool clComediIO::PcmStart (int *iSubDev*, double * *dpSampleRate*, int *iChannels*, unsigned int * *uipScanList*, unsigned int *uiListLength*, bool *bDither*, bool *bDoubleClock*) `[private]`

dpSampleRate = 1.0 / ((double) sComediCmd.convert_arg * iChannels / 1.0e9);

Definition at line 38 of file ComediIO.cc.

References comediDev, and SetError().

Referenced by PcmInStart().

7.22.4.3 long clComediIO::PcmRead (sampl_t **, long) `[private]`

Definition at line 86 of file ComediIO.cc.

References comediDev, and InBuf.

Referenced by PcmInRead().

7.22.4.4 bool clComediIO::Open (const char *)

Open device.

Parameters:
cpDevice Device filename

Returns:
Success?

Definition at line 122 of file ComediIO.cc.

References comediDev, and SetError().

Referenced by clComediSrv::AudioInThread().

7.22.4.5 bool clComediIO::Close ()

Close device.

Returns:
Success?

Definition at line 134 of file ComediIO.cc.

References comediDev, cpErrorMsg, iInSubDev, and SetError().

Referenced by clComediSrv::AudioInThread(), and ~clComediIO().

7.22.4.6 int clComediIO::GetSubdeviceCount ()

Get number of subdevices.

Returns:
Number of subdevices

Definition at line 163 of file ComediIO.cc.

References comediDev.

7.22.4.7 int clComediIO::GetVersionCode ()

Get version code.

Returns:
Version code

Definition at line 169 of file ComediIO.cc.

References comediDev, and SetError().

Referenced by clComediSrv::AudioInThread().

7.22.4.8 const char * clComediIO::GetDriverName ()

Get driver name.

Returns:
Driver name

Definition at line 180 of file ComediIO.cc.

References comediDev.

Referenced by clComediSrv::AudioInThread().

7.22.4.9 const char * clComediIO::GetBoardName ()

Get board name.

Returns:
Board name

Definition at line 186 of file ComediIO.cc.

References comediDev.

Referenced by clComediSrv::AudioInThread().

7.22.4.10 const char* clComediIO::GetErrorMsg () [inline]

Get error message.

Returns:
Error message

Definition at line 106 of file ComediIO.hh.

References cpErrorMsg.

Referenced by clComediSrv::AudioInThread().

7.22.4.11 bool clComediIO::PcmInOpen ()

Open PCM input device.

Returns:
Success?

Definition at line 192 of file ComediIO.cc.

References comediDev, iInMaxValue, iInSubDev, and SetError().

7.22.4.12 bool clComediIO::PcmInClose ()

Close PCM input device.

Returns:
Success?

Definition at line 216 of file ComediIO.cc.

References comediDev, iInSubDev, and SetError().

Referenced by clComediSrv::AudioInThread().

7.22.4.13 int clComediIO::PcmInGetChannelCount ()

Get number of channels for PCM input device.

Returns:
Number of channels

Definition at line 231 of file ComediIO.cc.

References comediDev, iInSubDev, and SetError().

7.22.4.14 bool clComediIO::PcmInStart (double *, int, double, bool = true, bool = false)

Start input operation.

Parameters:

dpSampleRate Samplerate (actual rate is returned)

iChannels Number of channels to use

dRange Input range (in volts)

bDither User dither?

bDoubleClock Use scan/channel timers instead of one?

Returns:

Success?

Definition at line 242 of file ComediIO.cc.

References comediDev, iInSubDev, InScanList, PcmStart(), and SetError().

Referenced by clComediSrv::AudioInThread().

7.22.4.15 bool clComediIO::PcmInStop ()

Stop input operation.

Returns:

Success?

Definition at line 265 of file ComediIO.cc.

References comediDev, iInSubDev, and SetError().

Referenced by clComediSrv::AudioInThread().

7.22.4.16 long clComediIO::PcmInRead (float *, long)

Read input samples.

Parameters:

Buffer

Buffer size (in samples)

Returns:

Number of samples

Definition at line 276 of file ComediIO.cc.

References iInMaxValue, PcmRead(), and SetError().

Referenced by clComediSrv::AudioInThread().

7.22.4.17 long clComediIO::PcmInRead (double *, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 296 of file ComediIO.cc.

References iInMaxValue, PcmRead(), and SetError().

7.22.4.18 bool clComediIO::PcmBufferSizeSet (unsigned *int*)

Set buffer size for PCM device.

Note:
Buffer size must be multiple of pagesize.

Parameters:
uiBufSize Buffer size

Returns:
Success?

Definition at line 316 of file ComediIO.cc.

References comediDev, iInSubDev, and SetError().

Referenced by clComediSrv::AudioInThread().

7.22.4.19 int clComediIO::PcmBufferSizeGet ()

Get buffer size for PCM device.

Returns:
Size of buffer

Definition at line 334 of file ComediIO.cc.

References comediDev, iInSubDev, and SetError().

Referenced by clComediSrv::AudioInThread().

7.22.4.20 size_t clComediIO::PcmGetIntSampleSize () `[inline]`

Get internal sample size (for calculating buffer sizes).

Returns:
Intenal sample size

Definition at line 174 of file ComediIO.hh.

Referenced by clComediSrv::AudioInThread().

7.22.5 Member Data Documentation

7.22.5.1 int clComediIO::iInSubDev [private]

Definition at line 38 of file ComediIO.hh.

Referenced by clComediIO(), Close(), PcmBufferSizeGet(), PcmBufferSizeSet(), PcmInClose(), PcmInGetChannelCount(), PcmInOpen(), PcmInStart(), and PcmInStop().

7.22.5.2 char* clComediIO::cpErrorMsg [private]

Definition at line 39 of file ComediIO.hh.

Referenced by clComediIO(), Close(), GetErrorMsg(), and SetError().

7.22.5.3 comedi_t* clComediIO::comediDev [private]

Definition at line 40 of file ComediIO.hh.

Referenced by clComediIO(), Close(), GetBoardName(), GetDriverName(), GetSubdeviceCount(), GetVersionCode(), Open(), PcmBufferSizeGet(), PcmBufferSizeSet(), PcmInClose(), PcmInGetChannelCount(), PcmInOpen(), PcmInStart(), PcmInStop(), PcmRead(), and PcmStart().

7.22.5.4 lsampl_t clComediIO::iInMaxValue [private]

Definition at line 41 of file ComediIO.hh.

Referenced by PcmInOpen(), and PcmInRead().

7.22.5.5 clAlloc clComediIO::InScanList [private]

Definition at line 42 of file ComediIO.hh.

Referenced by PcmInStart().

7.22.5.6 clAlloc clComediIO::InBuf [private]

Definition at line 43 of file ComediIO.hh.

Referenced by PcmRead().

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

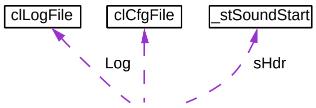
- **ComediIO.hh**
- **ComediIO.cc**

7.23 clComediSrv Class Reference

ComediServer.

`#include <ComediSrv.hh>`

Collaboration diagram for clComediSrv:



Public Member Functions

- **clComediSrv** ()
- ∼**clComediSrv** ()
- int **Main** (int, char ∗∗)
- void ∗ **AudioInThread** (void ∗)
- void ∗ **ServeClientThread** (void ∗)
- void **Stop** ()

Public Attributes

- **clLogFile Log**

Private Member Functions

- bool **GetAudioCfg** (char ∗, int ∗, double ∗, int ∗, double ∗, int ∗)
- bool **InitCompress** (int, int, int, int, int)

Private Attributes

- volatile bool **bRun**
- volatile int **iAudioBufSize**
- volatile int **iBlockCntr**
- **stSoundStart sHdr**
- clAlloc **AudioBuf**
- clAlloc **CompHead**
- clAlloc **FLACFrame**
- clMutex **MtxAudio**
- clCondition **CndAudio**
- **clCfgFile Cfg**

7.23.1 Detailed Description

ComediServer.

Input server with support for Comedi DAQ cards.

Definition at line 56 of file ComediSrv.hh.

7.23.2 Constructor & Destructor Documentation

7.23.2.1 clComediSrv::clComediSrv ()

Definition at line 253 of file ComediSrv.cc.

References clLogFile::Add(), bRun, Cfg, COM_CFGFILE, COM_LOGFILE, iAudioBufSize, iBlockCntr, Log, clLogFile::Open(), and clCfgFile::SetFileName().

7.23.2.2 clComediSrv::~clComediSrv ()

Definition at line 267 of file ComediSrv.cc.

References clLogFile::Add(), and Log.

7.23.3 Member Function Documentation

7.23.3.1 bool clComediSrv::GetAudioCfg (char *, int *, double *, int *, double *, int *) `[private]`

Definition at line 112 of file ComediSrv.cc.

References clLogFile::Add(), Cfg, clCfgFile::GetFlt(), clCfgFile::GetInt(), clCfgFile::GetStr(), and Log.

Referenced by AudioInThread().

7.23.3.2 bool clComediSrv::InitCompress (int, int, int, int, int) `[private]`

Definition at line 149 of file ComediSrv.cc.

References clLogFile::Add(), Cfg, FLACFrame, clCfgFile::GetInt(), Log, and MSG_SOUND_COMPRESS_FLAC.

Referenced by AudioInThread().

7.23.3.3 int clComediSrv::Main (int, char **)

Definition at line 280 of file ComediSrv.cc.

References clLogFile::Add(), bDaemon, clSockServ::Bind(), bRun, Cfg, COM_CONNECT_TIMEOUT, COM_SHUTDOWNFILE, clCfgFile::GetInt(), GLOBAL_VERSMAJ, GLOBAL_VERSMIN, GLOBAL_VERSPL, Log, SigHandler(), Stop(), clSockServ::WaitForConnect(), WrapAudioInThread(), and WrapServeClientThread().

Referenced by main().

7.23.3.4 void * clComediSrv::AudioInThread (void *)

Definition at line 356 of file ComediSrv.cc.

References clLogFile::Add(), AudioBuf, bRun, Cfg, clComediIO::Close(), CndAudio, COM_FRAG_SIZE_DEFAULT, COM_INTHREAD_PRIORITY, COM_LOGENTRY_SIZE, COM_USE_DITHER, _stSoundStart::dSampleRate, FLACFrame, GDT, GetAudioCfg(), clComediIO::GetBoardName(), clComediIO::GetDriverName(), clComediIO::GetErrorMsg(), clCfgFile::GetInt(), clComediIO::GetVersionCode(), iAudioBufSize, iBlockCntr, _stSoundStart::iChannels, _stSoundStart::iCompress, _stSoundStart::iFragmentSize, InitCompress(), Log, MSG_SOUND_COMPRESS_FLAC, MSG_SOUND_COMPRESS_NONE, MtxAudio, clComediIO::Open(), clComediIO::PcmBufferSizeGet(), clComediIO::PcmBufferSizeSet(), clComediIO::PcmGetIntSampleSize(), clComediIO::PcmInClose(), clComediIO::PcmInRead(), clComediIO::PcmInStart(), clComediIO::PcmInStop(), clSoundMsg::SetData(), sHdr, Stop(), clComediIO::VERS_MASK_MAJ, clComediIO::VERS_MASK_MIN, clComediIO::VERS_MASK_PL, clComediIO::VERS_SHIFT_MAJ, clComediIO::VERS_SHIFT_MIN, and clComediIO::VERS_SHIFT_PL.

Referenced by WrapAudioInThread().

7.23.3.5 void * clComediSrv::ServeClientThread (void *)

Definition at line 538 of file ComediSrv.cc.

References clLogFile::Add(), AudioBuf, bRun, Cfg, CndAudio, COM_LOGENTRY_SIZE, COM_OUTTHREAD_PRIORITY, COM_SOCKET_BUF_FRAGS, CompHead, clSockOp::DisableNagle(), clCfgFile::GetInt(), clSockOp::GetPeerName(), GLOBAL_HEADER_LEN, iAudioBufSize, iBlockCntr, _stSoundStart::iCompress, Log, MtxAudio, clSockOp::SetSendBufSize(), clSoundMsg::SetStart(), clSockOp::SetTypeOfService(), sHdr, SOp, and clSockOp::WriteN().

Referenced by WrapServeClientThread().

7.23.3.6 void clComediSrv::Stop () `[inline]`

Definition at line 86 of file ComediSrv.hh.

References bRun.

Referenced by AudioInThread(), Main(), and SigHandler().

7.23.4 Member Data Documentation

7.23.4.1 volatile bool clComediSrv::bRun [private]

Definition at line 58 of file ComediSrv.hh.

Referenced by AudioInThread(), clComediSrv(), Main(), ServeClientThread(), and Stop().

7.23.4.2 volatile int clComediSrv::iAudioBufSize [private]

Definition at line 59 of file ComediSrv.hh.

Referenced by AudioInThread(), clComediSrv(), and ServeClientThread().

7.23.4.3 volatile int clComediSrv::iBlockCntr [private]

Definition at line 60 of file ComediSrv.hh.

Referenced by AudioInThread(), clComediSrv(), and ServeClientThread().

7.23.4.4 stSoundStart clComediSrv::sHdr [private]

Definition at line 61 of file ComediSrv.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.23.4.5 clAlloc clComediSrv::AudioBuf [private]

Definition at line 65 of file ComediSrv.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.23.4.6 clAlloc clComediSrv::CompHead [private]

Definition at line 66 of file ComediSrv.hh.

Referenced by ServeClientThread().

7.23.4.7 clAlloc clComediSrv::FLACFrame [private]

Definition at line 67 of file ComediSrv.hh.

Referenced by AudioInThread(), and InitCompress().

7.23.4.8 clMutex clComediSrv::MtxAudio [private]

Definition at line 68 of file ComediSrv.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.23.4.9 clCondition clComediSrv::CndAudio [private]

Definition at line 72 of file ComediSrv.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.23.4.10 clCfgFile clComediSrv::Cfg [private]

Definition at line 73 of file ComediSrv.hh.

Referenced by AudioInThread(), clComediSrv(), GetAudioCfg(), InitCompress(), Main(), and ServeClientThread().

7.23.4.11 clLogFile clComediSrv::Log

Definition at line 80 of file ComediSrv.hh.

Referenced by AudioInThread(), clComediSrv(), GetAudioCfg(), InitCompress(), Main(), ServeClientThread(), SigHandler(), and ~clComediSrv().

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

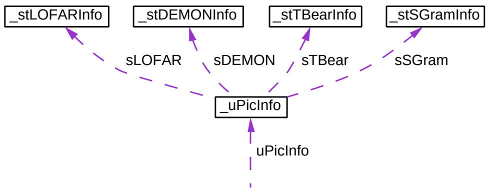
- **ComediSrv.hh**
- **ComediSrv.cc**

7.24 clConvPic Class Reference

Conversion of .tif + .tif.inf to image containing information in the .inf file.

`#include <ConvPic.hh>`

Collaboration diagram for clConvPic:



Public Member Functions

- **clConvPic** ()
- ~**clConvPic** ()
- int **Main** (int ∗, char ∗∗∗)

Private Member Functions

- bool **Initialize** ()
- bool **LoadLOFARInfo** (const char ∗)
- bool **LoadDEMONInfo** (const char ∗)
- bool **LoadSGramInfo** (const char ∗)
- bool **LoadTBearInfo** (const char ∗)
- bool **LoadAndCreate** (const char ∗)
- bool **CopySource** ()
- bool **DrawLOFARInfo** ()
- bool **DrawDEMONInfo** ()
- bool **DrawSGramInfo** ()
- bool **DrawTBearInfo** ()
- bool **ConvertColors** ()
- bool **SaveResult** (const char ∗)
- void **Clean** ()
- void **SplitLine** (const char ∗, string &, string &)
- time_t **ParseTime** (const char ∗)
- double **RadToDeg** (double)
- void **GetTextSize** (const char ∗, int ∗, int ∗, int ∗=NULL)

Private Attributes

- bool **bTwoColor**
- int **iType**
- int **iSrcWidth**
- int **iSrcHeight**
- int **iDstWidth**
- int **iDstHeight**
- char * **cpPageSize**
- **utPicInfo uPicInfo**
- Image * **ImgSrc**
- Image * **ImgDst**

7.24.1 Detailed Description

Conversion of .tif + .tif.inf to image containing information in the .inf file.

Definition at line 122 of file ConvPic.hh.

7.24.2 Constructor & Destructor Documentation

7.24.2.1 clConvPic::clConvPic ()

Definition at line 950 of file ConvPic.cc.

References bTwoColor, cpPageSize, and iType.

7.24.2.2 clConvPic::~clConvPic ()

Definition at line 958 of file ConvPic.cc.

7.24.3 Member Function Documentation

7.24.3.1 bool clConvPic::Initialize () `[private]`

Definition at line 43 of file ConvPic.cc.

Referenced by Main().

7.24.3.2 bool clConvPic::LoadLOFARInfo (const char *) `[private]`

Definition at line 49 of file ConvPic.cc.

References _stLOFARInfo::dHighFreq, _stLOFARInfo::dLineTime, _stLOFARInfo::dLowFreq, ParseTime(), _uPicInfo::sLOFAR, SplitLine(), _stLOFARInfo::ttTime, and uPicInfo.

Referenced by Main().

7.24.3.3 bool clConvPic::LoadDEMONInfo (const char *) [private]

Definition at line 85 of file ConvPic.cc.

References _stDEMONInfo::dDEMONBand, _stDEMONInfo::dHighFreq, _stDEMONInfo::dLineTime, _stDEMONInfo::dLowFreq, ParseTime(), _uPicInfo::sDEMON, SplitLine(), _stDEMONInfo::ttTime, and uPicInfo.

Referenced by Main().

7.24.3.4 bool clConvPic::LoadSGramInfo (const char *) [private]

Definition at line 123 of file ConvPic.cc.

References _stSGramInfo::dHighFreq, _stSGramInfo::dLineTime, _stSGramInfo::dLowFreq, ParseTime(), SplitLine(), _uPicInfo::sSGram, _stSGramInfo::ttTime, and uPicInfo.

Referenced by Main().

7.24.3.5 bool clConvPic::LoadTBearInfo (const char *) [private]

Definition at line 159 of file ConvPic.cc.

References _stTBearInfo::dHighFreq, _stTBearInfo::dIntTime, _stTBearInfo::dLeftDir, _stTBearInfo::dRightDir, _stTBearInfo::iSectors, ParseTime(), SplitLine(), _uPicInfo::sTBear, _stTBearInfo::ttTime, and uPicInfo.

Referenced by Main().

7.24.3.6 bool clConvPic::LoadAndCreate (const char *) [private]

Definition at line 199 of file ConvPic.cc.

References bTwoColor, CP_FONT_NAME, CP_MARGIN_B, CP_MARGIN_L, CP_MARGIN_R, CP_MARGIN_T, CP_PAGE_SPEC, cpPageSize, iDstHeight, iDstWidth, ImgDst, ImgSrc, iSrcHeight, and iSrcWidth.

Referenced by Main().

7.24.3.7 bool clConvPic::CopySource () [private]

Definition at line 270 of file ConvPic.cc.

References CP_MARGIN_L, CP_MARGIN_T, ImgDst, ImgSrc, iSrcHeight, and iSrcWidth.

Referenced by Main().

7.24.3.8 bool clConvPic::DrawLOFARInfo () [private]

Definition at line 291 of file ConvPic.cc.

References CP_FONT_LARGE, CP_FONT_NORMAL, CP_FONT_SMALL, CP_MARGIN_L, CP_MARGIN_T, CP_TICKL_LEN, CP_TICKS_LEN, _stLOFARInfo::dHighFreq, _stLOFARInfo::dLineTime, _stLOFARInfo::dLowFreq, GetTextSize(), ImgDst, iSrcHeight, iSrcWidth, _uPicInfo::sLOFAR, _stLOFARInfo::ttTime, and uPicInfo.

Referenced by Main().

7.24.3.9 bool clConvPic::DrawDEMONInfo () [private]

Definition at line 417 of file ConvPic.cc.

References CP_FONT_LARGE, CP_FONT_NORMAL, CP_FONT_SMALL, CP_MARGIN_L, CP_MARGIN_T, CP_TICKL_LEN, CP_TICKS_LEN, _stDEMONInfo::dDEMONBand, _stDEMONInfo::dHighFreq, _stDEMONInfo::dLineTime, _stDEMONInfo::dLowFreq, GetTextSize(), ImgDst, iSrcHeight, iSrcWidth, _uPicInfo::sDEMON, _stDEMONInfo::ttTime, and uPicInfo.

Referenced by Main().

7.24.3.10 bool clConvPic::DrawSGramInfo () [private]

Definition at line 543 of file ConvPic.cc.

References CP_FONT_LARGE, CP_FONT_NORMAL, CP_FONT_SMALL, CP_MARGIN_L, CP_MARGIN_T, CP_TICKL_LEN, CP_TICKS_LEN, _stSGramInfo::dHighFreq, _stSGramInfo::dLineTime, _stSGramInfo::dLowFreq, GetTextSize(), ImgDst, iSrcHeight, iSrcWidth, _uPicInfo::sSGram, _stSGramInfo::ttTime, and uPicInfo.

Referenced by Main().

7.24.3.11 bool clConvPic::DrawTBearInfo () [private]

Definition at line 683 of file ConvPic.cc.

References CP_FONT_LARGE, CP_FONT_NORMAL, CP_FONT_SMALL, CP_MARGIN_L, CP_MARGIN_T, CP_TICKL_LEN, CP_TICKS_LEN, _stTBearInfo::dIntTime, _stTBearInfo::dLeftDir, _stTBearInfo::dRightDir, GetTextSize(), ImgDst, _stTBearInfo::iSectors, iSrcHeight, iSrcWidth, RadToDeg(), _uPicInfo::sTBear, _stTBearInfo::ttTime, and uPicInfo.

Referenced by Main().

7.24.3.12 bool clConvPic::ConvertColors () [private]

Definition at line 804 of file ConvPic.cc.

References ImgDst.

Referenced by Main().

7.24.3.13 bool clConvPic::SaveResult (const char *) [private]

Definition at line 841 of file ConvPic.cc.

References cpPageSize, and ImgDst.

Referenced by Main().

7.24.3.14 void clConvPic::Clean () [private]

Definition at line 864 of file ConvPic.cc.

References ImgDst, and ImgSrc.

Referenced by Main().

7.24.3.15 void clConvPic::SplitLine (const char *, string &, string &) [private]

Definition at line 878 of file ConvPic.cc.

Referenced by LoadDEMONInfo(), LoadLOFARInfo(), LoadSGramInfo(), and LoadTBearInfo().

7.24.3.16 time_t clConvPic::ParseTime (const char *) [private]

Definition at line 895 of file ConvPic.cc.

Referenced by LoadDEMONInfo(), LoadLOFARInfo(), LoadSGramInfo(), and LoadTBearInfo().

7.24.3.17 double clConvPic::RadToDeg (double) [inline, private]

Definition at line 923 of file ConvPic.cc.

Referenced by DrawTBearInfo().

7.24.3.18 void clConvPic::GetTextSize (const char *, int *, int *, int * = NULL) [private]

Definition at line 929 of file ConvPic.cc.

References ImgDst.

Referenced by DrawDEMONInfo(), DrawLOFARInfo(), DrawSGramInfo(), and DrawTBearInfo().

7.24.3.19 int clConvPic::Main (int *, char ***)

Definition at line 963 of file ConvPic.cc.

References bTwoColor, Clean(), ConvertColors(), CopySource(), CP_TYPE_DEMON, CP_TYPE_LOFAR, CP_TYPE_SGRAM, CP_TYPE_TBEAR, cpPageSize, DrawDEMONInfo(), DrawLOFARInfo(), DrawSGramInfo(), DrawTBearInfo(), Initialize(), iType, LoadAndCreate(), LoadDEMONInfo(), LoadLOFARInfo(), LoadSGramInfo(), LoadTBearInfo(), and SaveResult().

Referenced by main().

7.24.4 Member Data Documentation

7.24.4.1 bool clConvPic::bTwoColor `[private]`

Definition at line 124 of file ConvPic.hh.

Referenced by clConvPic(), LoadAndCreate(), and Main().

7.24.4.2 int clConvPic::iType `[private]`

Definition at line 125 of file ConvPic.hh.

Referenced by clConvPic(), and Main().

7.24.4.3 int clConvPic::iSrcWidth `[private]`

Definition at line 126 of file ConvPic.hh.

Referenced by CopySource(), DrawDEMONInfo(), DrawLOFARInfo(), DrawSGramInfo(), DrawTBearInfo(), and LoadAndCreate().

7.24.4.4 int clConvPic::iSrcHeight `[private]`

Definition at line 127 of file ConvPic.hh.

Referenced by CopySource(), DrawDEMONInfo(), DrawLOFARInfo(), DrawSGramInfo(), DrawTBearInfo(), and LoadAndCreate().

7.24.4.5 int clConvPic::iDstWidth `[private]`

Definition at line 128 of file ConvPic.hh.

Referenced by LoadAndCreate().

7.24.4.6 int clConvPic::iDstHeight `[private]`

Definition at line 129 of file ConvPic.hh.

Referenced by LoadAndCreate().

7.24.4.7 char* clConvPic::cpPageSize [private]

Definition at line 130 of file ConvPic.hh.

Referenced by clConvPic(), LoadAndCreate(), Main(), and SaveResult().

7.24.4.8 utPicInfo clConvPic::uPicInfo [private]

Definition at line 131 of file ConvPic.hh.

Referenced by DrawDEMONInfo(), DrawLOFARInfo(), DrawSGramInfo(), DrawTBearInfo(), LoadDEMONInfo(), LoadLOFARInfo(), LoadSGramInfo(), and LoadTBearInfo().

7.24.4.9 Image* clConvPic::ImgSrc [private]

Definition at line 132 of file ConvPic.hh.

Referenced by Clean(), CopySource(), and LoadAndCreate().

7.24.4.10 Image* clConvPic::ImgDst [private]

Definition at line 133 of file ConvPic.hh.

Referenced by Clean(), ConvertColors(), CopySource(), DrawDEMONInfo(), DrawLOFARInfo(), DrawSGramInfo(), DrawTBearInfo(), GetTextSize(), LoadAndCreate(), and SaveResult().

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

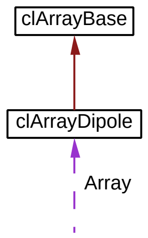
- **ConvPic.hh**
- **ConvPic.cc**

7.25 clCorrDipole Class Reference

Correlator class for dipole array.

`#include <CorrDipole.hh>`

Collaboration diagram for clCorrDipole:



Public Member Functions

- **clCorrDipole** (**clArrayDipole** ∗, GDT, int, long, bool, bool)

    Constructor.

- ∼**clCorrDipole** ()
- bool **AddData** ()

    Add data to integration buffers.

- GDT **Process** (GDT)

    Process data in integration buffers and return result.

- void **SetHistory** ()

    Set history part of buffer.

- GDT **GetPeakLevel** ()

    Get peak level (in dB) of integration buffer.

- GDT **GetIntegrationTime** ()

    Get real integration time.

Private Attributes

- bool **bDebug**
- bool **bFilter**
- long **lSampleCount**

    Sample count for specified integration time.

- long **lBaseIdx**

  Starting index = maximum delay.

- long **lWinSize**

  Window size (input data).

- GDT **fIntTime**

  Integration time in seconds.

- clDSPAlloc **Data** [2]

  Data buffers for integration.

- clDSPOp **DSP**
- clDSPOp **DSPBank** [2]
- **clArrayDipole** ∗ **Array**

7.25.1 Detailed Description

Correlator class for dipole array.

Definition at line 36 of file CorrDipole.hh.

7.25.2 Constructor & Destructor Documentation

7.25.2.1 clCorrDipole::clCorrDipole (clArrayDipole ∗, GDT, int, long, bool, bool)

Constructor.

Parameters:

ArrayPtr Pointer to **clArrayDipole**(p. 25)

fIntegrationTime Integration time (s)

iSampleRate Samplerate

lWindowSize Size of window

bDisableFilter Disable source data filtering

bEnableDebug Enable debugging output

Definition at line 31 of file CorrDipole.cc.

References Array, bDebug, bFilter, Data, DSPBank, fIntTime, GDT, clArray-Dipole::GetMaxDelay(), lBaseIdx, lSampleCount, and lWinSize.

7.25.2.2 clCorrDipole::∼clCorrDipole ()

Definition at line 63 of file CorrDipole.cc.

7.25.3 Member Function Documentation

7.25.3.1 bool clCorrDipole::AddData ()

Add data to integration buffers.

Call this after calling **clArrayDipole::AddData()**(p. 31) Returns >= 1 when integration buffers are full and ready to be processed by **Process()**(p. 160), call until returns 0 before calling **clArrayDipole::AddData()**(p. 31) again.

Definition at line 68 of file CorrDipole.cc.

References Array, bFilter, Data, DSPBank, GDT, clArrayDipole::GetFiltPtr(), clArrayDipole::GetRawPtr(), lBaseIdx, lSampleCount, and lWinSize.

Referenced by clDirection::ProcessData().

7.25.3.2 GDT clCorrDipole::Process (GDT)

Process data in integration buffers and return result.

This can be called many times for different directions.

Parameters:

fDirection Direction (rad)

Returns:

Normalized cross-correlation

Definition at line 93 of file CorrDipole.cc.

References Array, Data, DSP, GDT, clArrayDipole::GetDelaySamples(), lBaseIdx, and lSampleCount.

Referenced by clDirection::CalcDipoleCorrs().

7.25.3.3 void clCorrDipole::SetHistory ()

Set history part of buffer.

Call this between last call of **Process()**(p. 160) and next call of **AddData()**(p. 160), this will copy data from end of integration buffer to start of integration buffer.

Definition at line 111 of file CorrDipole.cc.

References Data, DSPBank, GDT, lBaseIdx, and lSampleCount.

Referenced by clDirection::CalcDipoleCorrs().

7.25.3.4 GDT clCorrDipole::GetPeakLevel ()

Get peak level (in dB) of integration buffer.

Returns:
Peak level (dB)

Definition at line 125 of file CorrDipole.cc.

References Data, DIR_DB_SCALE, DSPBank, GDT, lBaseIdx, and lSample-Count.

Referenced by clDirection::ProcessData().

7.25.3.5 GDT clCorrDipole::GetIntegrationTime () [inline]

Get real integration time.

Returns:
Integration time (s)

Definition at line 98 of file CorrDipole.hh.

References fIntTime, and GDT.

7.25.4 Member Data Documentation

7.25.4.1 bool clCorrDipole::bDebug [private]

Definition at line 38 of file CorrDipole.hh.

Referenced by clCorrDipole().

7.25.4.2 bool clCorrDipole::bFilter [private]

Definition at line 39 of file CorrDipole.hh.

Referenced by AddData(), and clCorrDipole().

7.25.4.3 long clCorrDipole::lSampleCount [private]

Sample count for specified integration time.

Definition at line 40 of file CorrDipole.hh.

Referenced by AddData(), clCorrDipole(), GetPeakLevel(), Process(), and Set-History().

7.25.4.4 long clCorrDipole::lBaseIdx [private]

Starting index = maximum delay.

Definition at line 41 of file CorrDipole.hh.

Referenced by AddData(), clCorrDipole(), GetPeakLevel(), Process(), and Set-History().

7.25.4.5 long clCorrDipole::lWinSize [private]

Window size (input data).

Definition at line 42 of file CorrDipole.hh.

Referenced by AddData(), and clCorrDipole().

7.25.4.6 GDT clCorrDipole::fIntTime [private]

Integration time in seconds.

Definition at line 43 of file CorrDipole.hh.

Referenced by clCorrDipole(), and GetIntegrationTime().

7.25.4.7 clDSPAlloc clCorrDipole::Data[2] [private]

Data buffers for integration.

Definition at line 44 of file CorrDipole.hh.

Referenced by AddData(), clCorrDipole(), GetPeakLevel(), Process(), and SetHistory().

7.25.4.8 clDSPOp clCorrDipole::DSP [private]

Definition at line 45 of file CorrDipole.hh.

Referenced by Process().

7.25.4.9 clDSPOp clCorrDipole::DSPBank[2] [private]

Definition at line 46 of file CorrDipole.hh.

Referenced by AddData(), clCorrDipole(), GetPeakLevel(), and SetHistory().

7.25.4.10 clArrayDipole* clCorrDipole::Array [private]

Definition at line 47 of file CorrDipole.hh.

Referenced by AddData(), clCorrDipole(), and Process().

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

- **CorrDipole.hh**
- **CorrDipole.cc**

7.26 _stDEMONInfo Struct Reference

Information stored in saved DEMON .tif.inf.

`#include <ConvPic.hh>`

Public Attributes

- time_t **ttTime**
  Oldest dataline time.
- double **dLowFreq**
  Lower frequency limit.
- double **dHighFreq**
  Upper frequency limit.
- double **dDEMONBand**
  DEMON bandwidth.
- double **dLineTime**
  Length of one scanline in time (s).

7.26.1 Detailed Description

Information stored in saved DEMON .tif.inf.

Definition at line 73 of file ConvPic.hh.

7.26.2 Member Data Documentation

7.26.2.1 time_t _stDEMONInfo::ttTime

Oldest dataline time.

Definition at line 75 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), and clConvPic::LoadDEMONInfo().

7.26.2.2 double _stDEMONInfo::dLowFreq

Lower frequency limit.

Definition at line 76 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), and clConvPic::LoadDEMONInfo().

7.26.2.3 double _stDEMONInfo::dHighFreq

Upper frequency limit.

Definition at line 77 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), and clConvPic::LoadDEMONInfo().

7.26.2.4 double _stDEMONInfo::dDEMONBand

DEMON bandwidth.

Definition at line 78 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), and clConvPic::LoadDEMONInfo().

7.26.2.5 double _stDEMONInfo::dLineTime

Length of one scanline in time (s).

Definition at line 79 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), and clConvPic::LoadDEMONInfo().

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

- **ConvPic.hh**

7.27 _stDirCfg Struct Reference

Configuration information message for subnodes.

`#include <Locate.hh>`

Public Attributes

- long **lWindowSize**
- float **fSoundSpeed**
- float **fLowFrequency**
- float **fIntegrationTime**
- int **iScaling**
- float **fScalingExp**
- int **iRemoveNoise**
- float **fAlpha**
- long **lMeanLength**
- long **lGapLength**

7.27.1 Detailed Description

Configuration information message for subnodes.

Definition at line 55 of file Locate.hh.

7.27.2 Member Data Documentation

7.27.2.1 long _stDirCfg::lWindowSize

Definition at line 57 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), clSubLocate::Initialize(), and clLocate::Initialize().

7.27.2.2 float _stDirCfg::fSoundSpeed

Definition at line 58 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), and clLocate::Initialize().

7.27.2.3 float _stDirCfg::fLowFrequency

Definition at line 59 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), and clLocate::Initialize().

7.27.2.4 float _stDirCfg::fIntegrationTime

Definition at line 60 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), and clLocate::Initialize().

7.27.2.5 int _stDirCfg::iScaling

Definition at line 61 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), and clLocate::Initialize().

7.27.2.6 float _stDirCfg::fScalingExp

Definition at line 62 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), and clLocate::Initialize().

7.27.2.7 int _stDirCfg::iRemoveNoise

Definition at line 63 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), and clLocate::Initialize().

7.27.2.8 float _stDirCfg::fAlpha

Definition at line 64 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), and clLocate::Initialize().

7.27.2.9 long _stDirCfg::lMeanLength

Definition at line 65 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), and clLocate::Initialize().

7.27.2.10 long _stDirCfg::lGapLength

Definition at line 66 of file Locate.hh.

Referenced by clSubLocate::ConnectDir(), and clLocate::Initialize().

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

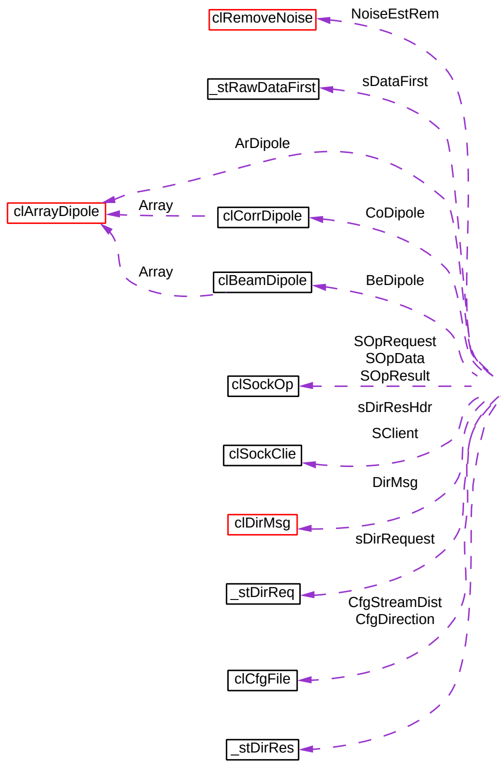
- **Locate.hh**

7.28 clDirection Class Reference

Direction calculation server.

`#include <Direction.hh>`

Collaboration diagram for clDirection:



Public Member Functions

- **clDirection** (int, int)
- ∼**clDirection** ()
- int **Exec** ()

  Main method, call this only.

- void ∗ **ReceiveData** (void ∗)

  Receive data thread.

- void ∗ **ProcessData** (void ∗)

  Dataprocessing thread (there can be multiple instances of these).

- void ∗ **SendResults** (void ∗)

Send results thread.

Private Member Functions

- bool **GetRequest** ()
- bool **GetFirstMsg** ()
- bool **InitArray** ()
- void **Stop** ()
- void **Scale** (GDT ∗)
- void **CalcDipoleBeams** (int, long, long, bool)
- void **CalcDipoleCorrs** (int, long, long, bool)
- void **StartTiming** ()
- void **StopTiming** ()

Private Attributes

- volatile bool **bRun**
- volatile bool **bConnected**
- int **iProcessorCount**
- int **iStartChannel**
- int **iArrayType**
- int **iShadingType**
- int **iHorizSensors**
- int **iVertSensors**
- int **iRawBufSize**
- volatile int **iResultsRefCount**
- long **lWindowSize**
- double **dTimingReal**
- GDT **fSensorSpacing**
- GDT **fpHorizSpacing** [BF_MAX_X_SENSORS]
- GDT **fpVertSpacing** [BF_MAX_Y_SENSORS]
- clock_t **ctTiming**
- pthread_t **tidReceiveData**
- pthread_t **tidpProcessData** [DIR_MAX_CPUS]
- pthread_t **tidSendResults**
- **stDirReq sDirRequest**
- **stDirRes sDirResHdr**
- **stRawDataFirst sDataFirst**
- clCondition **CndDataReady**
- clCondition **CndResultsReady**
- clMutex **MtxClassData**
- clMutex **MtxDataReady**
- clMutex **MtxResultsReady**
- **clCfgFile** ∗ **CfgStreamDist**
- **clCfgFile** ∗ **CfgDirection**

- **clSockClie SClient**
- **clSockOp * SOpRequest**
- **clSockOp * SOpResult**
- **clSockOp * SOpData**
- **clDirMsg DirMsg**
- clDSPAlloc **RawData**
- clDSPAlloc **DirResData**
- clDSPOp **DSP**
- **clRemoveNoise NoiseEstRem**
- **clArrayDipole ArDipole**
- **clBeamDipole * BeDipole** [DIR_MAX_CPUS]
- **clCorrDipole * CoDipole** [DIR_MAX_CPUS]

7.28.1 Detailed Description

Direction calculation server.

Definition at line 67 of file Direction.hh.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 clDirection::clDirection (int, int)

Definition at line 390 of file Direction.cc.

References bConnected, BeDipole, bRun, CfgDirection, CfgStreamDist, CoDipole, DIR_CFGFILE, DIR_DEF_WIN_SIZE, DIR_MAX_CPUS, iProcessorCount, iResultsRefCount, iStartChannel, lWindowSize, SD_CFGFILE, SOpData, SOpRequest, and SOpResult.

7.28.2.2 clDirection::~clDirection ()

Definition at line 413 of file Direction.cc.

References BeDipole, CfgDirection, CfgStreamDist, CoDipole, iProcessorCount, SOpData, SOpRequest, and SOpResult.

7.28.3 Member Function Documentation

7.28.3.1 bool clDirection::GetRequest () `[private]`

Definition at line 106 of file Direction.cc.

References bDebug, DIR_REQ_TIMEOUT, DirMsg, clSockOp::GetErrno(), clDirMsg::GetRequest(), GLOBAL_HEADER_LEN, clSockOp::ReadN(), clSockOp::ReadSelect(), sDirRequest, and SOpRequest.

Referenced by Exec().

7.28.3.2 bool clDirection::GetFirstMsg () [private]

Definition at line 126 of file Direction.cc.

References bDebug, DIR_RAW1ST_TIMEOUT, clSockOp::GetErrno(), _stRawDataReq::iChannel, clSockOp::ReadN(), clSockOp::ReadSelect(), sDataFirst, SOpData, and clSockOp::WriteN().

Referenced by Exec().

7.28.3.3 bool clDirection::InitArray () [private]

Definition at line 153 of file Direction.cc.

References ArDipole, bDaemon, bDebug, BeDipole, CfgDirection, cpaArrayTypes, DIR_ARRAY_TYPE_DIPOLE, DIR_ARRAY_TYPE_LINE, DIR_ARRAY_TYPE_TRIANGLE, _stRawDataFirst::dSampleRate, _stDirRes::fIntegrationTime, _stDirReq::fIntegrationTime, _stDirReq::fLowFreqLimit, fpHorizSpacing, fpVertSpacing, fSensorSpacing, _stDirReq::fSoundSpeed, GDT, clCfgFile::GetFlt(), clCfgFile::GetFltArray(), clCfgFile::GetInt(), clBeamDipole::GetIntegrationTime(), _stDirReq::iAlgorithm, iArrayType, _stRawDataFirst::iChannels, iHorizSensors, clArrayDipole::Initialize(), iProcessorCount, iRawBufSize, iShadingType, iVertSensors, lWindowSize, MSG_DIR_ALG_BEAM, sDataFirst, sDirRequest, and sDirResHdr.

Referenced by Exec().

7.28.3.4 void clDirection::Stop () [private]

Definition at line 260 of file Direction.cc.

References bRun, and MtxClassData.

Referenced by ReceiveData(), and SendResults().

7.28.3.5 void clDirection::Scale (GDT ∗) [inline, private]

Definition at line 268 of file Direction.cc.

References _stDirReq::fScalingExp, GDT, _stDirReq::iScaling, MSG_DIR_SCAL_EXP, MSG_DIR_SCAL_LIN, MSG_DIR_SCAL_LOG, MSG_DIR_SCAL_SIN, and sDirRequest.

Referenced by CalcDipoleBeams(), and CalcDipoleCorrs().

7.28.3.6 void clDirection::CalcDipoleBeams (int, long, long, bool) [private]

Definition at line 293 of file Direction.cc.

References BeDipole, DirResData, _stDirReq::fLeftDir, _stDirReq::fRightDir, GDT, _stDirReq::lSectorCount, MtxClassData, clBeamDipole::Process(), Scale(), sDirRequest, and clBeamDipole::SetHistory().

Referenced by ProcessData().

7.28.3.7 void clDirection::CalcDipoleCorrs (int, long, long, bool) [private]

Definition at line 322 of file Direction.cc.

References CoDipole, DirResData, _stDirReq::fLeftDir, _stDirReq::fRightDir, GDT, _stDirReq::lSectorCount, MtxClassData, clCorrDipole::Process(), Scale(), sDirRequest, and clCorrDipole::SetHistory().

Referenced by ProcessData().

7.28.3.8 void clDirection::StartTiming () [private]

Definition at line 350 of file Direction.cc.

References ctTiming, and dTimingReal.

Referenced by ProcessData().

7.28.3.9 void clDirection::StopTiming () [private]

Definition at line 361 of file Direction.cc.

References bDebug, ctTiming, dTimingReal, _stDirRes::fIntegrationTime, and sDirResHdr.

Referenced by ProcessData().

7.28.3.10 int clDirection::Exec ()

Main method, call this only.

Definition at line 430 of file Direction.cc.

References bDaemon, bDebug, CfgDirection, CfgStreamDist, clSockClie::Connect(), DirResData, GDT, GetFirstMsg(), clCfgFile::GetInt(), GetRequest(), clCfgFile::GetStr(), InitArray(), iProcessorCount, iRawBufSize, _stDirReq::iRemoveNoise, iStartChannel, _stDirReq::lSectorCount, lWindowSize, MSG_DIR_BNER_OTA, MSG_DIR_BNER_TPSW, MtxClassData, RawData, SClient, sDirRequest, SOpData, ThrProcessData(), ThrReceiveData(), ThrSendResults(), tidpProcessData, tidReceiveData, and tidSendResults.

Referenced by main().

7.28.3.11 void * clDirection::ReceiveData (void *)

Receive data thread.

Definition at line 512 of file Direction.cc.

References clArrayDipole::AddData(), ArDipole, bDebug, bRun, CndDataReady, DIR_ARRAY_TYPE_CYLINDER, DIR_ARRAY_TYPE_DIPOLE, DIR_ARRAY_TYPE_LINE, DIR_ARRAY_TYPE_PLANE, DIR_ARRAY_TYPE_SPHERE, DIR_ARRAY_TYPE_TRIANGLE, DIR_TIMEOUT, clSockOp::GetErrno(), iArrayType, _stRawDataFirst::iChannels, iRawBufSize, iStartChannel, MtxClassData, MtxDataReady, RawData, clSockOp::ReadN(), clSockOp::ReadSelect(), sDataFirst, SOpData, and Stop().

Referenced by ThrReceiveData().

7.28.3.12 void ∗ clDirection::ProcessData (void ∗)

Dataprocessing thread (there can be multiple instances of these).

Definition at line 559 of file Direction.cc.

References clCorrDipole::AddData(), clBeamDipole::AddData(), bDebug, BeDipole, bRun, CalcDipoleBeams(), CalcDipoleCorrs(), CndDataReady, CndResultsReady, CoDipole, DIR_ARRAY_TYPE_CYLINDER, DIR_ARRAY_TYPE_DIPOLE, DIR_ARRAY_TYPE_LINE, DIR_ARRAY_TYPE_PLANE, DIR_ARRAY_TYPE_SPHERE, DIR_ARRAY_TYPE_TRIANGLE, _stDirRes::fPeakLevel, clCorrDipole::GetPeakLevel(), clBeamDipole::GetPeakLevel(), _stDirReq::iAlgorithm, iArrayType, iProcessorCount, iResultsRefCount, _stDirReq::lSectorCount, MSG_DIR_ALG_BEAM, MSG_DIR_ALG_CORR, MtxClassData, MtxDataReady, MtxResultsReady, sDirRequest, sDirResHdr, StartTiming(), StopTiming(), and tidpProcessData.

Referenced by ThrProcessData().

7.28.3.13 void ∗ clDirection::SendResults (void ∗)

Send results thread.

Definition at line 681 of file Direction.cc.

References bDebug, _stDirReq::bNormalize, bRun, CndResultsReady, DIR_TIMEOUT, DirMsg, DirResData, DSP, _stDirReq::fAlpha, GDT, clSockOp::GetErrno(), GLOBAL_HEADER_LEN, iProcessorCount, _stDirReq::iRemoveNoise, iResultsRefCount, _stDirReq::lGapLength, _stDirReq::lMeanLength, _stDirReq::lSectorCount, MSG_DIR_BNER_OTA, MSG_DIR_BNER_TPSW, MtxClassData, MtxResultsReady, NoiseEstRem, clRemoveNoise::OTA(), sDirRequest, sDirResHdr, clDirMsg::SetResult(), SOpResult, _stDirRes::sTimeStamp, Stop(), clRemoveNoise::TPSW(), clSockOp::WriteN(), and clSockOp::WriteSelect().

Referenced by ThrSendResults().

7.28.4 Member Data Documentation

7.28.4.1 volatile bool clDirection::bRun [private]

Definition at line 70 of file Direction.hh.

Referenced by clDirection(), ProcessData(), ReceiveData(), SendResults(), and Stop().

7.28.4.2 volatile bool clDirection::bConnected [private]

Definition at line 71 of file Direction.hh.

Referenced by clDirection().

7.28.4.3 int clDirection::iProcessorCount [private]

Definition at line 72 of file Direction.hh.

Referenced by clDirection(), Exec(), InitArray(), ProcessData(), SendResults(), and ∼clDirection().

7.28.4.4 int clDirection::iStartChannel [private]

Definition at line 73 of file Direction.hh.

Referenced by clDirection(), Exec(), and ReceiveData().

7.28.4.5 int clDirection::iArrayType [private]

Definition at line 74 of file Direction.hh.

Referenced by InitArray(), ProcessData(), and ReceiveData().

7.28.4.6 int clDirection::iShadingType [private]

Definition at line 75 of file Direction.hh.

Referenced by InitArray().

7.28.4.7 int clDirection::iHorizSensors [private]

Definition at line 76 of file Direction.hh.

Referenced by InitArray().

7.28.4.8 int clDirection::iVertSensors [private]

Definition at line 77 of file Direction.hh.

Referenced by InitArray().

7.28.4.9 int clDirection::iRawBufSize [private]

Definition at line 78 of file Direction.hh.

Referenced by Exec(), InitArray(), and ReceiveData().

7.28.4.10 volatile int clDirection::iResultsRefCount [private]

Definition at line 79 of file Direction.hh.

Referenced by clDirection(), ProcessData(), and SendResults().

7.28.4.11 long clDirection::lWindowSize [private]

Definition at line 80 of file Direction.hh.

Referenced by clDirection(), Exec(), and InitArray().

7.28.4.12 double clDirection::dTimingReal [private]

Definition at line 81 of file Direction.hh.

Referenced by StartTiming(), and StopTiming().

7.28.4.13 GDT clDirection::fSensorSpacing [private]

Definition at line 82 of file Direction.hh.

Referenced by InitArray().

7.28.4.14 GDT clDirection::fpHorizSpacing[BF_MAX_X_SENSORS] [private]

Definition at line 83 of file Direction.hh.

Referenced by InitArray().

7.28.4.15 GDT clDirection::fpVertSpacing[BF_MAX_Y_SENSORS] [private]

Definition at line 84 of file Direction.hh.

Referenced by InitArray().

7.28.4.16 clock_t clDirection::ctTiming [private]

Definition at line 85 of file Direction.hh.

Referenced by StartTiming(), and StopTiming().

7.28.4.17 pthread_t clDirection::tidReceiveData [private]

Definition at line 86 of file Direction.hh.

Referenced by Exec().

7.28.4.18 pthread_t clDirection::tidpProcessData[DIR_MAX_CPUS] [private]

Definition at line 87 of file Direction.hh.

Referenced by Exec(), and ProcessData().

7.28.4.19 pthread_t clDirection::tidSendResults [private]

Definition at line 88 of file Direction.hh.

Referenced by Exec().

7.28.4.20 stDirReq clDirection::sDirRequest [private]

Definition at line 90 of file Direction.hh.

Referenced by CalcDipoleBeams(), CalcDipoleCorrs(), Exec(), GetRequest(), InitArray(), ProcessData(), Scale(), and SendResults().

7.28.4.21 stDirRes clDirection::sDirResHdr [private]

Definition at line 91 of file Direction.hh.

Referenced by InitArray(), ProcessData(), SendResults(), and StopTiming().

7.28.4.22 stRawDataFirst clDirection::sDataFirst [private]

Definition at line 92 of file Direction.hh.

Referenced by GetFirstMsg(), InitArray(), and ReceiveData().

7.28.4.23 clCondition clDirection::CndDataReady [private]

Definition at line 94 of file Direction.hh.

Referenced by ProcessData(), and ReceiveData().

7.28.4.24 clCondition clDirection::CndResultsReady [private]

Definition at line 95 of file Direction.hh.

Referenced by ProcessData(), and SendResults().

7.28.4.25 clMutex clDirection::MtxClassData [private]

Definition at line 96 of file Direction.hh.

Referenced by CalcDipoleBeams(), CalcDipoleCorrs(), Exec(), ProcessData(), ReceiveData(), SendResults(), and Stop().

7.28.4.26 clMutex clDirection::MtxDataReady [private]

Definition at line 97 of file Direction.hh.

Referenced by ProcessData(), and ReceiveData().

7.28.4.27 clMutex clDirection::MtxResultsReady [private]

Definition at line 98 of file Direction.hh.

Referenced by ProcessData(), and SendResults().

7.28.4.28 clCfgFile* clDirection::CfgStreamDist [private]

Definition at line 99 of file Direction.hh.

Referenced by clDirection(), Exec(), and ∼clDirection().

7.28.4.29 clCfgFile* clDirection::CfgDirection [private]

Definition at line 100 of file Direction.hh.

Referenced by clDirection(), Exec(), InitArray(), and ∼clDirection().

7.28.4.30 clSockClie clDirection::SClient [private]

Definition at line 101 of file Direction.hh.

Referenced by Exec().

7.28.4.31 clSockOp* clDirection::SOpRequest [private]

Definition at line 102 of file Direction.hh.

Referenced by clDirection(), GetRequest(), and ∼clDirection().

7.28.4.32 clSockOp* clDirection::SOpResult [private]

Definition at line 103 of file Direction.hh.

Referenced by clDirection(), SendResults(), and ~clDirection().

7.28.4.33 clSockOp* clDirection::SOpData [private]

Definition at line 104 of file Direction.hh.

Referenced by clDirection(), Exec(), GetFirstMsg(), ReceiveData(), and ~cl-Direction().

7.28.4.34 clDirMsg clDirection::DirMsg [private]

Definition at line 105 of file Direction.hh.

Referenced by GetRequest(), and SendResults().

7.28.4.35 clDSPAlloc clDirection::RawData [private]

Definition at line 106 of file Direction.hh.

Referenced by Exec(), and ReceiveData().

7.28.4.36 clDSPAlloc clDirection::DirResData [private]

Definition at line 107 of file Direction.hh.

Referenced by CalcDipoleBeams(), CalcDipoleCorrs(), Exec(), and Send-Results().

7.28.4.37 clDSPOp clDirection::DSP [private]

Definition at line 108 of file Direction.hh.

Referenced by SendResults().

7.28.4.38 clRemoveNoise clDirection::NoiseEstRem [private]

Definition at line 109 of file Direction.hh.

Referenced by SendResults().

7.28.4.39 clArrayDipole clDirection::ArDipole [private]

Definition at line 110 of file Direction.hh.

Referenced by InitArray(), and ReceiveData().

7.28.4.40 clBeamDipole* clDirection::BeDipole[DIR_MAX_CPUS] [private]

Definition at line 111 of file Direction.hh.

Referenced by CalcDipoleBeams(), clDirection(), InitArray(), ProcessData(), and ~clDirection().

7.28.4.41 clCorrDipole* clDirection::CoDipole[DIR_MAX_CPUS] [private]

Definition at line 112 of file Direction.hh.

Referenced by CalcDipoleCorrs(), clDirection(), ProcessData(), and ~clDirection().

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

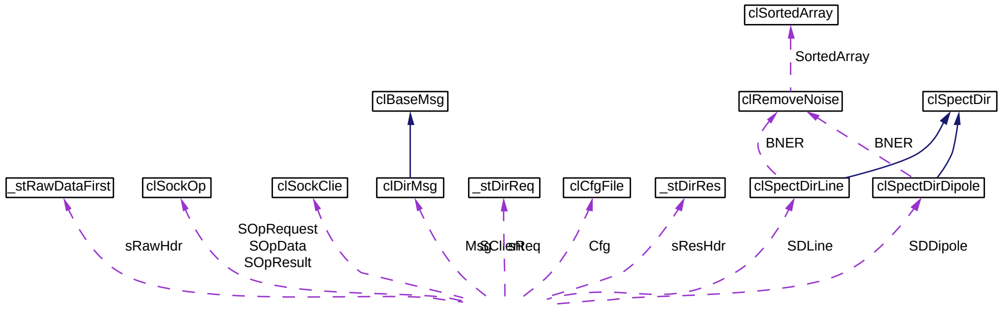
- **Direction.hh**
- **Direction.cc**

7.29 clDirection2 Class Reference

Spectrum based direction server.

`#include <Direction2.hh>`

Collaboration diagram for clDirection2:



Public Member Functions

- **clDirection2** (int, int)
- **∼clDirection2** ()
- int **Main** ()

Private Member Functions

- bool **GetCfg** ()
- bool **GetRq** ()
- bool **ConnectStream** ()
- bool **InitDir** ()
- void **ProcessLoop** ()

Private Attributes

- bool **bRun**
- int **iFilterType**
- int **iArrayType**
- long **lSensorCount**
- long **lFilterSize**
- long **lWindowSize**
- long **lStartChannel**
- GDT **fSensorSpacing**
- char **cpStreamSocket** [_POSIX_PATH_MAX+1]
- **stDirReq sReq**

- **stDirRes sResHdr**
- **stRawDataFirst sRawHdr**
- **clCfgFile Cfg**
- clDSPOp **DSP**
- **clDirMsg Msg**
- **clSockClie SClient**
- **clSockOp SOpRequest**
- **clSockOp SOpResult**
- **clSockOp SOpData**
- **clSpectDirDipole ∗ SDDipole**
- **clSpectDirLine ∗ SDLine**

7.29.1 Detailed Description

Spectrum based direction server.

Definition at line 58 of file Direction2.hh.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 clDirection2::clDirection2 (int, int)

Definition at line 288 of file Direction2.cc.

References bRun, clSockOp::SetHandle(), SOpRequest, and SOpResult.

7.29.2.2 clDirection2::∼clDirection2 ()

Definition at line 296 of file Direction2.cc.

7.29.3 Member Function Documentation

7.29.3.1 bool clDirection2::GetCfg () `[private]`

Definition at line 83 of file Direction2.cc.

References Cfg, cpStreamSocket, DIR2_CFGFILE, DIR2_DEF_FFT_SIZE, DIR2_DEF_FILT_SIZE, fSensorSpacing, clCfgFile::GetFlt(), clCfgFile::GetInt(), clCfgFile::GetStr(), iArrayType, iFilterType, lFilterSize, lSensorCount, lStartChannel, lWindowSize, SD_CFGFILE, and clCfgFile::SetFileName().

Referenced by Main().

7.29.3.2 bool clDirection2::GetRq () `[private]`

Definition at line 103 of file Direction2.cc.

References DIR2_REQ_TIMEOUT, clDirMsg::GetRequest(), GLOBAL_HEADER_LEN, Msg, clSockOp::ReadN(), clSockOp::ReadSelect(), SOpRequest, and sReq.

Referenced by Main().

7.29.3.3 bool clDirection2::ConnectStream () [private]

Definition at line 115 of file Direction2.cc.

References clSockClie::Connect(), cpStreamSocket, DIR2_RAW1ST_TIMEOUT, _stRawDataReq::iChannel, clSockOp::ReadN(), clSockOp::ReadSelect(), SClient, clSockOp::SetHandle(), SOpData, sRawHdr, and clSockOp::WriteN().

Referenced by Main().

7.29.3.4 bool clDirection2::InitDir () [private]

Definition at line 133 of file Direction2.cc.

References bDebug, DIR2_ARRAY_TYPE_CYLINDER, DIR2_ARRAY_TYPE_DIPOLE, DIR2_ARRAY_TYPE_LINE, DIR2_ARRAY_TYPE_PLANE, DIR2_ARRAY_TYPE_SPHERE, DIR2_ARRAY_TYPE_TRIANGLE, _stRawDataFirst::dSampleRate, _stDirReq::fIntegrationTime, fSensorSpacing, _stDirReq::fSoundSpeed, _stDirReq::iAlgorithm, iArrayType, iFilterType, lFilterSize, _stDirReq::lSectorCount, lSensorCount, lWindowSize, MSG_DIR_ALG_SPECT, SDDipole, SDLine, sRawHdr, and sReq.

Referenced by Main().

7.29.3.5 void clDirection2::ProcessLoop () [private]

Definition at line 205 of file Direction2.cc.

References _stDirRes::b3DArray, bDebug, bRun, DIR2_ARRAY_TYPE_CYLINDER, DIR2_ARRAY_TYPE_DIPOLE, DIR2_ARRAY_TYPE_LINE, DIR2_ARRAY_TYPE_PLANE, DIR2_ARRAY_TYPE_SPHERE, DIR2_ARRAY_TYPE_TRIANGLE, DSP, _stDirReq::fAlpha, _stSpectDirRN::fAlpha, _stDirRes::fHighFreqLimit, _stDirRes::fIntegrationTime, _stDirReq::fLowFreqLimit, _stDirRes::fPeakLevel, GDT, clSpectDirLine::GetIntegrationTime(), clSpectDirDipole::GetIntegrationTime(), clSpectDirLine::GetResults(), clSpectDirDipole::GetResults(), GLOBAL_HEADER_LEN, iArrayType, _stRawDataFirst::iChannels, _stDirReq::iRemoveNoise, _stDirReq::iScaling, _stSpectDirRN::iType, lFilterSize, _stDirReq::lGapLength, _stSpectDirRN::lGapLength, _stDirReq::lMeanLength, _stSpectDirRN::lMeanLength, _stDirRes::lSectorCount, _stDirReq::lSectorCount, lStartChannel, Msg, clSpectDirLine::PutData(), clSpectDirDipole::PutData(), clSockOp::ReadN(), clSpectDirLine::ResetResults(), clSpectDirDipole::ResetResults(), SDDipole, SDLine, clSockOp::SetRecvBufSize(), clDirMsg::SetResult(), clSockOp::Set-

SendBufSize(), SOpData, SOpResult, sRawHdr, sReq, sResHdr, _stDirRes::s-TimeStamp, and clSockOp::WriteN().

Referenced by Main().

7.29.3.6 int clDirection2::Main ()

Definition at line 301 of file Direction2.cc.

References bDebug, ConnectStream(), GetCfg(), GetRq(), InitDir(), and ProcessLoop().

Referenced by main().

7.29.4 Member Data Documentation

7.29.4.1 bool clDirection2::bRun [private]

Definition at line 60 of file Direction2.hh.

Referenced by clDirection2(), and ProcessLoop().

7.29.4.2 int clDirection2::iFilterType [private]

Definition at line 61 of file Direction2.hh.

Referenced by GetCfg(), and InitDir().

7.29.4.3 int clDirection2::iArrayType [private]

Definition at line 62 of file Direction2.hh.

Referenced by GetCfg(), InitDir(), and ProcessLoop().

7.29.4.4 long clDirection2::lSensorCount [private]

Definition at line 63 of file Direction2.hh.

Referenced by GetCfg(), and InitDir().

7.29.4.5 long clDirection2::lFilterSize [private]

Definition at line 64 of file Direction2.hh.

Referenced by GetCfg(), InitDir(), and ProcessLoop().

7.29.4.6 long clDirection2::lWindowSize [private]

Definition at line 65 of file Direction2.hh.

Referenced by GetCfg(), and InitDir().

7.29.4.7 long clDirection2::lStartChannel [private]

Definition at line 66 of file Direction2.hh.

Referenced by GetCfg(), and ProcessLoop().

7.29.4.8 GDT clDirection2::fSensorSpacing [private]

Definition at line 67 of file Direction2.hh.

Referenced by GetCfg(), and InitDir().

7.29.4.9 char clDirection2::cpStreamSocket[_POSIX_PATH_MAX + 1] [private]

Definition at line 68 of file Direction2.hh.

Referenced by ConnectStream(), and GetCfg().

7.29.4.10 stDirReq clDirection2::sReq [private]

Definition at line 69 of file Direction2.hh.

Referenced by GetRq(), InitDir(), and ProcessLoop().

7.29.4.11 stDirRes clDirection2::sResHdr [private]

Definition at line 70 of file Direction2.hh.

Referenced by ProcessLoop().

7.29.4.12 stRawDataFirst clDirection2::sRawHdr [private]

Definition at line 71 of file Direction2.hh.

Referenced by ConnectStream(), InitDir(), and ProcessLoop().

7.29.4.13 clCfgFile clDirection2::Cfg [private]

Definition at line 72 of file Direction2.hh.

Referenced by GetCfg().

7.29.4.14 clDSPOp clDirection2::DSP [private]

Definition at line 73 of file Direction2.hh.

Referenced by ProcessLoop().

7.29.4.15 clDirMsg clDirection2::Msg [private]

Definition at line 74 of file Direction2.hh.

Referenced by GetRq(), and ProcessLoop().

7.29.4.16 clSockClie clDirection2::SClient [private]

Definition at line 75 of file Direction2.hh.

Referenced by ConnectStream().

7.29.4.17 clSockOp clDirection2::SOpRequest [private]

Definition at line 76 of file Direction2.hh.

Referenced by clDirection2(), and GetRq().

7.29.4.18 clSockOp clDirection2::SOpResult [private]

Definition at line 77 of file Direction2.hh.

Referenced by clDirection2(), and ProcessLoop().

7.29.4.19 clSockOp clDirection2::SOpData [private]

Definition at line 78 of file Direction2.hh.

Referenced by ConnectStream(), and ProcessLoop().

7.29.4.20 clSpectDirDipole* clDirection2::SDDipole [private]

Definition at line 79 of file Direction2.hh.

Referenced by InitDir(), and ProcessLoop().

7.29.4.21 clSpectDirLine* clDirection2::SDLine [private]

Definition at line 80 of file Direction2.hh.

Referenced by InitDir(), and ProcessLoop().

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

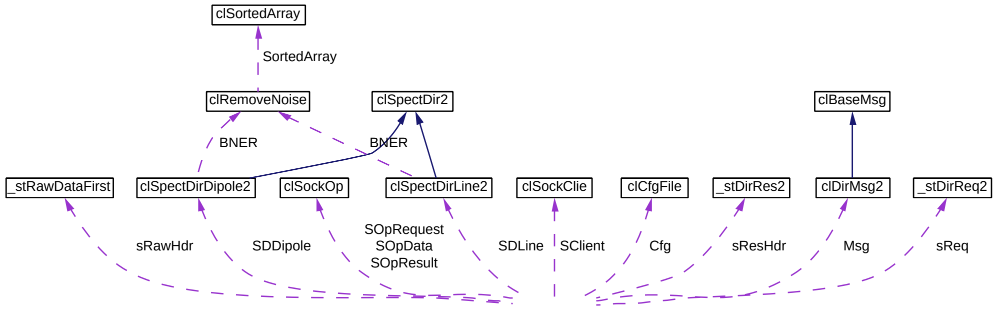
- **Direction2.hh**
- **Direction2.cc**

7.30 clDirection3 Class Reference

Spectrum based direction server.

`#include <Direction3.hh>`

Collaboration diagram for clDirection3:



Public Member Functions

- **clDirection3** (int, int)
- **~clDirection3** ()
- int **Main** ()

Private Member Functions

- bool **GetCfg** ()
- bool **GetRq** ()
- bool **ConnectStream** ()
- bool **InitDir** ()
- void **ProcessLoop** ()

Private Attributes

- bool **bRun**
- int **iFilterType**
- int **iArrayType**
- long **lSensorCount**
- long **lFilterSize**
- long **lWindowSize**
- long **lStartChannel**
- GDT **fSensorSpacing**
- char **cpStreamSocket** [_POSIX_PATH_MAX+1]
- **stDirReq2 sReq**

- **stDirRes2 sResHdr**
- **stRawDataFirst sRawHdr**
- **clCfgFile Cfg**
- clDSPOp **DSP**
- **clDirMsg2 Msg**
- **clSockClie SClient**
- **clSockOp SOpRequest**
- **clSockOp SOpResult**
- **clSockOp SOpData**
- **clSpectDirDipole2 * SDDipole**
- **clSpectDirLine2 * SDLine**

7.30.1 Detailed Description

Spectrum based direction server.

Definition at line 58 of file Direction3.hh.

7.30.2 Constructor & Destructor Documentation

7.30.2.1 clDirection3::clDirection3 (int, int)

Definition at line 284 of file Direction3.cc.

References bRun, clSockOp::SetHandle(), SOpRequest, and SOpResult.

7.30.2.2 clDirection3::~clDirection3 ()

Definition at line 292 of file Direction3.cc.

7.30.3 Member Function Documentation

7.30.3.1 bool clDirection3::GetCfg () `[private]`

Definition at line 83 of file Direction3.cc.

References Cfg, cpStreamSocket, DIR3_CFGFILE, DIR3_DEF_FILT_SIZE, fSensorSpacing, clCfgFile::GetFlt(), clCfgFile::GetInt(), clCfgFile::GetStr(), iArrayType, iFilterType, lFilterSize, lSensorCount, lStartChannel, SD_-CFGFILE, and clCfgFile::SetFileName().

Referenced by Main().

7.30.3.2 bool clDirection3::GetRq () `[private]`

Definition at line 103 of file Direction3.cc.

References DIR3_REQ_TIMEOUT, clDirMsg2::GetRequest(), GLOBAL_HEADER_LEN, Msg, clSockOp::ReadN(), clSockOp::ReadSelect(), SOpRequest, and sReq.

Referenced by Main().

7.30.3.3 bool clDirection3::ConnectStream () [private]

Definition at line 115 of file Direction3.cc.

References clSockClie::Connect(), cpStreamSocket, DIR3_RAW1ST_TIMEOUT, _stRawDataReq::iChannel, clSockOp::ReadN(), clSockOp::ReadSelect(), SClient, clSockOp::SetHandle(), SOpData, sRawHdr, and clSockOp::WriteN().

Referenced by Main().

7.30.3.4 bool clDirection3::InitDir () [private]

Definition at line 133 of file Direction3.cc.

References bDebug, DIR3_ARRAY_TYPE_CYLINDER, DIR3_ARRAY_TYPE_DIPOLE, DIR3_ARRAY_TYPE_LINE, DIR3_ARRAY_TYPE_PLANE, DIR3_ARRAY_TYPE_SPHERE, DIR3_ARRAY_TYPE_TRIANGLE, _stRawDataFirst::dSampleRate, _stDirReq2::fIntegrationTime, fSensorSpacing, _stDirReq2::fSoundSpeed, iArrayType, iFilterType, lFilterSize, lSensorCount, _stDirReq2::lWindowSize, SDDipole, SDLine, sRawHdr, and sReq.

Referenced by Main().

7.30.3.5 void clDirection3::ProcessLoop () [private]

Definition at line 191 of file Direction3.cc.

References bDebug, bRun, DIR3_ARRAY_TYPE_CYLINDER, DIR3_ARRAY_TYPE_DIPOLE, DIR3_ARRAY_TYPE_LINE, DIR3_ARRAY_TYPE_PLANE, DIR3_ARRAY_TYPE_SPHERE, DIR3_ARRAY_TYPE_TRIANGLE, DSP, _stDirReq2::fAlpha, _stSpectDir2RN::fAlpha, _stDirRes2::fFreqResolution, _stDirRes2::fIntegrationTime, _stDirReq2::fLowFreqLimit, _stDirRes2::fPeakLevel, GDT, clSpectDirLine2::GetDirections(), clSpectDirDipole2::GetDirections(), clSpectDirLine2::GetFreqResolution(), clSpectDirDipole2::GetFreqResolution(), clSpectDirLine2::GetIntegrationTime(), clSpectDirDipole2::GetIntegrationTime(), clSpectDirLine2::GetLevels(), clSpectDirDipole2::GetLevels(), clSpectDirLine2::GetMaxBin(), clSpectDirDipole2::GetMaxBin(), clSpectDirLine2::GetMinBin(), clSpectDirDipole2::GetMinBin(), clSpectDirLine2::GetResultCount(), clSpectDirDipole2::GetResultCount(), clSpectDirLine2::GetResults(), clSpectDirDipole2::GetResults(), GLOBAL_HEADER_LEN, iArrayType, _stRawDataFirst::iChannels, _stDirReq2::iRemoveNoise, _stDirReq2::iScaling, _stSpectDir2RN::iType, lFilterSize, _stDirReq2::lGapLength, _stSpectDir2RN::lGapLength, _stDirRes2::lMaxBin, _stDirReq2::lMeanLength, _stSpectDir2RN::lMeanLength, _stDirRes2::lMinBin, _stDirRes2::lResultCount, lStartChannel, Msg, clSpectDirLine2::PutData(), clSpectDirDipole2::PutData(), clSockOp::ReadN(), SDDipole, SDLine, clSockOp::SetRecvBufSize(), clDirMsg2::SetResult(), clSockOp::SetSendBufSize(), SOpData, SOpResult, sRawHdr, sReq, sResHdr, _stDirRes2::sTimeStamp, and clSockOp::WriteN().

Referenced by Main().

7.30.3.6 int clDirection3::Main ()

Definition at line 297 of file Direction3.cc.

References bDebug, ConnectStream(), GetCfg(), GetRq(), InitDir(), and ProcessLoop().

Referenced by main().

7.30.4 Member Data Documentation

7.30.4.1 bool clDirection3::bRun `[private]`

Definition at line 60 of file Direction3.hh.

Referenced by clDirection3(), and ProcessLoop().

7.30.4.2 int clDirection3::iFilterType `[private]`

Definition at line 61 of file Direction3.hh.

Referenced by GetCfg(), and InitDir().

7.30.4.3 int clDirection3::iArrayType `[private]`

Definition at line 62 of file Direction3.hh.

Referenced by GetCfg(), InitDir(), and ProcessLoop().

7.30.4.4 long clDirection3::lSensorCount `[private]`

Definition at line 63 of file Direction3.hh.

Referenced by GetCfg(), and InitDir().

7.30.4.5 long clDirection3::lFilterSize `[private]`

Definition at line 64 of file Direction3.hh.

Referenced by GetCfg(), InitDir(), and ProcessLoop().

7.30.4.6 long clDirection3::lWindowSize [private]

Definition at line 65 of file Direction3.hh.

7.30.4.7 long clDirection3::lStartChannel [private]

Definition at line 66 of file Direction3.hh.

Referenced by GetCfg(), and ProcessLoop().

7.30.4.8 GDT clDirection3::fSensorSpacing [private]

Definition at line 67 of file Direction3.hh.

Referenced by GetCfg(), and InitDir().

7.30.4.9 char clDirection3::cpStreamSocket[_POSIX_PATH_MAX + 1] [private]

Definition at line 68 of file Direction3.hh.

Referenced by ConnectStream(), and GetCfg().

7.30.4.10 stDirReq2 clDirection3::sReq [private]

Definition at line 69 of file Direction3.hh.

Referenced by GetRq(), InitDir(), and ProcessLoop().

7.30.4.11 stDirRes2 clDirection3::sResHdr [private]

Definition at line 70 of file Direction3.hh.

Referenced by ProcessLoop().

7.30.4.12 stRawDataFirst clDirection3::sRawHdr [private]

Definition at line 71 of file Direction3.hh.

Referenced by ConnectStream(), InitDir(), and ProcessLoop().

7.30.4.13 clCfgFile clDirection3::Cfg [private]

Definition at line 72 of file Direction3.hh.

Referenced by GetCfg().

7.30.4.14 clDSPOp clDirection3::DSP [private]

Definition at line 73 of file Direction3.hh.

Referenced by ProcessLoop().

7.30.4.15 clDirMsg2 clDirection3::Msg [private]

Definition at line 74 of file Direction3.hh.

Referenced by GetRq(), and ProcessLoop().

7.30.4.16 clSockClie clDirection3::SClient [private]

Definition at line 75 of file Direction3.hh.

Referenced by ConnectStream().

7.30.4.17 clSockOp clDirection3::SOpRequest [private]

Definition at line 76 of file Direction3.hh.

Referenced by clDirection3(), and GetRq().

7.30.4.18 clSockOp clDirection3::SOpResult [private]

Definition at line 77 of file Direction3.hh.

Referenced by clDirection3(), and ProcessLoop().

7.30.4.19 clSockOp clDirection3::SOpData [private]

Definition at line 78 of file Direction3.hh.

Referenced by ConnectStream(), and ProcessLoop().

7.30.4.20 clSpectDirDipole2* clDirection3::SDDipole [private]

Definition at line 79 of file Direction3.hh.

Referenced by InitDir(), and ProcessLoop().

7.30.4.21 clSpectDirLine2* clDirection3::SDLine [private]

Definition at line 80 of file Direction3.hh.

Referenced by InitDir(), and ProcessLoop().

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

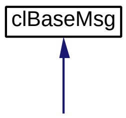
- **Direction3.hh**
- **Direction3.cc**

7.31 clDirMsg Class Reference

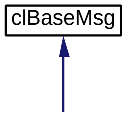
Direction server communication.

#include <Messages.hh>

Inheritance diagram for clDirMsg:



Collaboration diagram for clDirMsg:



Public Member Functions

- void **SetRequest** (char ∗, const **stpDirReq**)
- void **GetRequest** (const char ∗, **stpDirReq**)
- void **SetResult** (void ∗, const **stpDirRes**, const float ∗)
- void **SetResult** (void ∗, const **stpDirRes**, const double ∗)
- void **GetResult** (const void ∗, **stpDirRes**, float ∗)
- void **GetResult** (const void ∗, **stpDirRes**, double ∗)

7.31.1 Detailed Description

Direction server communication.

Definition at line 515 of file Messages.hh.

7.31.2 Member Function Documentation

7.31.2.1 void clDirMsg::SetRequest (char ∗, const *stpDirReq*)

Definition at line 466 of file Messages.cc.

Referenced by main(), and clGUIDir::SendSettings().

7.31.2.2 void clDirMsg::GetRequest (const char *, stpDirReq)

Definition at line 513 of file Messages.cc.

References _stDirReq::bDisableFilter, _stDirReq::bNormalize, _stDirReq::f-Alpha, _stDirReq::fIntegrationTime, _stDirReq::fLeftDir, _stDirReq::fLow-FreqLimit, _stDirReq::fRightDir, _stDirReq::fScalingExp, _stDirReq::fSound-Speed, _stDirReq::iAlgorithm, _stDirReq::iRemoveNoise, _stDirReq::iScaling, _-stDirReq::lGapLength, _stDirReq::lMeanLength, _stDirReq::lSectorCount, and MSG_TRUE.

Referenced by clDirection::GetRequest(), and clDirection2::GetRq().

7.31.2.3 void clDirMsg::SetResult (void *, const *stpDirRes*, const float *)

Definition at line 539 of file Messages.cc.

References _stDirRes::b3DArray, clBaseMsg::EndianConv(), _stDirRes::fHigh-FreqLimit, _stDirRes::fIntegrationTime, _stDirRes::fPeakLevel, GLOBAL_-HEADER_LEN, _stDirRes::lSectorCount, MSG_FALSE, MSG_TRUE, and _st-DirRes::sTimeStamp.

Referenced by clDirection2::ProcessLoop(), and clDirection::SendResults().

7.31.2.4 void clDirMsg::SetResult (void *, const *stpDirRes*, const double *)

Definition at line 575 of file Messages.cc.

References _stDirRes::b3DArray, clBaseMsg::EndianConv(), _stDirRes::fHigh-FreqLimit, _stDirRes::fIntegrationTime, _stDirRes::fPeakLevel, GLOBAL_-HEADER_LEN, _stDirRes::lSectorCount, MSG_FALSE, MSG_TRUE, and _st-DirRes::sTimeStamp.

7.31.2.5 void clDirMsg::GetResult (const void *, stpDirRes, float *)

Definition at line 611 of file Messages.cc.

References _stDirRes::b3DArray, clBaseMsg::EndianConv(), _stDirRes::fHigh-FreqLimit, _stDirRes::fIntegrationTime, _stDirRes::fPeakLevel, GLOBAL_-HEADER_LEN, _stDirRes::lSectorCount, MSG_TRUE, and _stDirRes::sTime-Stamp.

Referenced by main(), and clGUIDir::OnGdkInput().

7.31.2.6 void clDirMsg::GetResult (const void *, stpDirRes, double *)

Definition at line 638 of file Messages.cc.

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

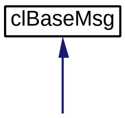
- **Messages.hh**
- **Messages.cc**

7.32 clDirMsg2 Class Reference

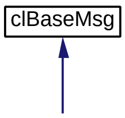
Direction server communication 2 (for locate/direction3).

#include <Messages.hh>

Inheritance diagram for clDirMsg2:



Collaboration diagram for clDirMsg2:



Public Member Functions

- void **SetRequest** (char ∗, const **stpDirReq2**)
- void **GetRequest** (const char ∗, **stpDirReq2**)
- void **SetResult** (void ∗, const **stpDirRes2**, const float ∗, const float ∗)
- void **SetResult** (void ∗, const **stpDirRes2**, const double ∗, const double ∗)
- void **GetResult** (const void ∗, **stpDirRes2**, float ∗, float ∗)
- void **GetResult** (const void ∗, **stpDirRes2**, double ∗, double ∗)

7.32.1 Detailed Description

Direction server communication 2 (for locate/direction3).

Definition at line 532 of file Messages.hh.

7.32.2 Member Function Documentation

7.32.2.1 void clDirMsg2::SetRequest (char ∗, const *stpDirReq2*)

Definition at line 665 of file Messages.cc.

Referenced by clSubLocate::ConnectDir().

7.32.2.2 void clDirMsg2::GetRequest (const char *, stpDirReq2)

Definition at line 697 of file Messages.cc.

References _stDirReq2::fAlpha, _stDirReq2::fIntegrationTime, _stDirReq2::fLowFreqLimit, _stDirReq2::fScalingExp, _stDirReq2::fSoundSpeed, _stDirReq2::iRemoveNoise, _stDirReq2::iScaling, _stDirReq2::lGapLength, _stDirReq2::lMeanLength, and _stDirReq2::lWindowSize.

Referenced by clDirection3::GetRq().

7.32.2.3 void clDirMsg2::SetResult (void *, const *stpDirRes2*, const float *, const float *)

Definition at line 714 of file Messages.cc.

References clBaseMsg::EndianConv(), _stDirRes2::fFreqResolution, _stDirRes2::fIntegrationTime, _stDirRes2::fPeakLevel, GLOBAL_HEADER_LEN, _stDirRes2::lMaxBin, _stDirRes2::lMinBin, _stDirRes2::lResultCount, and _stDirRes2::sTimeStamp.

Referenced by clDirection3::ProcessLoop().

7.32.2.4 void clDirMsg2::SetResult (void *, const *stpDirRes2*, const double *, const double *)

Definition at line 753 of file Messages.cc.

References clBaseMsg::EndianConv(), _stDirRes2::fFreqResolution, _stDirRes2::fIntegrationTime, _stDirRes2::fPeakLevel, GLOBAL_HEADER_LEN, _stDirRes2::lMaxBin, _stDirRes2::lMinBin, _stDirRes2::lResultCount, and _stDirRes2::sTimeStamp.

7.32.2.5 void clDirMsg2::GetResult (const void *, stpDirRes2, float *, float *)

Definition at line 792 of file Messages.cc.

References clBaseMsg::EndianConv(), _stDirRes2::fFreqResolution, _stDirRes2::fIntegrationTime, _stDirRes2::fPeakLevel, GLOBAL_HEADER_LEN, _stDirRes2::lMaxBin, _stDirRes2::lMinBin, _stDirRes2::lResultCount, and _stDirRes2::sTimeStamp.

Referenced by clSubLocate::Main().

7.32.2.6 void clDirMsg2::GetResult (const void *, stpDirRes2, double *, double *)

Definition at line 823 of file Messages.cc.

References clBaseMsg::EndianConv(), _stDirRes2::fFreqResolution, _stDirRes2::fIntegrationTime, _stDirRes2::fPeakLevel, GLOBAL_HEADER_LEN, _stDirRes2::lMaxBin, _stDirRes2::lMinBin, _stDirRes2::lResultCount, and _stDirRes2::sTimeStamp.

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

- **Messages.hh**
- **Messages.cc**

7.33 _stDirReq Struct Reference

Direction: request.

`#include <Messages.hh>`

Public Attributes

- int **iAlgorithm**
  Algorithm.
- float **fSoundSpeed**
  Speed of sound (m/s).
- float **fLowFreqLimit**
  Lower frequency limit (Hz).
- float **fIntegrationTime**
  Itegration time (s).
- int **iScaling**
  Scaling function.
- float **fScalingExp**
  Exponent for exponential scaling.
- bool **bNormalize**
  Normalization.
- float **fLeftDir**
  Left direction (rad).
- float **fRightDir**
  Right direction (rad).
- long **lSectorCount**
  Number of sectors.
- int **iRemoveNoise**
  Background noise estimation and removal algorithm.
- float **fAlpha**
  Alpha for noise removal.
- long **lMeanLength**
  Mean length for noise removal.

- long **lGapLength**

  Gap length for TPSW noise removal algorithm.

- bool **bDisableFilter**

  Disable all filtering of input signal.

7.33.1 Detailed Description

Direction: request.

Definition at line 250 of file Messages.hh.

7.33.2 Member Data Documentation

7.33.2.1 int _stDirReq::iAlgorithm

Algorithm.

Definition at line 252 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), clDirection::InitArray(), clGUIDir::InitConnection(), clDirection2::InitDir(), main(), and clDirection::ProcessData().

7.33.2.2 float _stDirReq::fSoundSpeed

Speed of sound (m/s).

Definition at line 253 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), clDirection::InitArray(), clDirection2::InitDir(), and main().

7.33.2.3 float _stDirReq::fLowFreqLimit

Lower frequency limit (Hz).

Definition at line 254 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), clDirection::InitArray(), main(), and clDirection2::ProcessLoop().

7.33.2.4 float _stDirReq::fIntegrationTime

Itegration time (s).

Definition at line 255 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), clDirection::InitArray(), clDirection2::InitDir(), and main().

7.33.2.5 int _stDirReq::iScaling

Scaling function.

Definition at line 256 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), main(), clDirection2::ProcessLoop(), and clDirection::Scale().

7.33.2.6 float _stDirReq::fScalingExp

Exponent for exponential scaling.

Definition at line 257 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), main(), and clDirection::Scale().

7.33.2.7 bool _stDirReq::bNormalize

Normalization.

Definition at line 258 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), main(), and clDirection::SendResults().

7.33.2.8 float _stDirReq::fLeftDir

Left direction (rad).

Definition at line 259 of file Messages.hh.

Referenced by clDirection::CalcDipoleBeams(), clDirection::CalcDipoleCorrs(), clDirMsg::GetRequest(), clGUIDir::GetSettings(), main(), and clGUIDir::SaveInfo().

7.33.2.9 float _stDirReq::fRightDir

Right direction (rad).

Definition at line 260 of file Messages.hh.

Referenced by clDirection::CalcDipoleBeams(), clDirection::CalcDipoleCorrs(), clDirMsg::GetRequest(), clGUIDir::GetSettings(), main(), and clGUIDir::SaveInfo().

7.33.2.10 long _stDirReq::lSectorCount

Number of sectors.

Definition at line 261 of file Messages.hh.

Referenced by clDirection::CalcDipoleBeams(), clDirection::CalcDipoleCorrs(), clDirection::Exec(), clDirMsg::GetRequest(), clGUIDir::GetSettings(), clDirection2::InitDir(), main(), clDirection::ProcessData(), clDirection2::ProcessLoop(), and clDirection::SendResults().

7.33.2.11 int _stDirReq::iRemoveNoise

Background noise estimation and removal algorithm.

Definition at line 262 of file Messages.hh.

Referenced by clDirection::Exec(), clDirMsg::GetRequest(), clGUIDir::GetSettings(), clDirection2::ProcessLoop(), and clDirection::SendResults().

7.33.2.12 float _stDirReq::fAlpha

Alpha for noise removal.

Definition at line 263 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), clDirection2::ProcessLoop(), and clDirection::SendResults().

7.33.2.13 long _stDirReq::lMeanLength

Mean length for noise removal.

Definition at line 264 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), clDirection2::ProcessLoop(), and clDirection::SendResults().

7.33.2.14 long _stDirReq::lGapLength

Gap length for TPSW noise removal algorithm.

Definition at line 265 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), clGUIDir::GetSettings(), clDirection2::ProcessLoop(), and clDirection::SendResults().

7.33.2.15 bool _stDirReq::bDisableFilter

Disable all filtering of input signal.

Definition at line 266 of file Messages.hh.

Referenced by clDirMsg::GetRequest(), and clGUIDir::GetSettings().

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

- **Messages.hh**

7.34 _stDirReq2 Struct Reference

Direction3: request.

`#include <Messages.hh>`

Public Attributes

- long **lWindowSize**
  Window size.
- float **fSoundSpeed**
  Speed of sound (m/s).
- float **fLowFreqLimit**
  Lower frequecy limit (Hz).
- float **fIntegrationTime**
  Integration time (s).
- int **iScaling**
  Scaling algorithm.
- float **fScalingExp**
  Exponent for exponential scaling.
- int **iRemoveNoise**
  Background noise estimation and removal algorithm.
- float **fAlpha**
  Alpha for noise removal.
- long **lMeanLength**
  Mean length for noise removal.
- long **lGapLength**
  Gap length for TPSW noise removal algorithm.

7.34.1 Detailed Description

Direction3: request.

This is used by locate server.

Definition at line 287 of file Messages.hh.

7.34.2 Member Data Documentation

7.34.2.1 long _stDirReq2::lWindowSize

Window size.

Definition at line 289 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), clDirMsg2::GetRequest(), and clDirection3::InitDir().

7.34.2.2 float _stDirReq2::fSoundSpeed

Speed of sound (m/s).

Definition at line 290 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), clDirMsg2::GetRequest(), and clDirection3::InitDir().

7.34.2.3 float _stDirReq2::fLowFreqLimit

Lower frequecy limit (Hz).

Definition at line 291 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), clDirMsg2::GetRequest(), and clDirection3::ProcessLoop().

7.34.2.4 float _stDirReq2::fIntegrationTime

Integration time (s).

Definition at line 292 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), clDirMsg2::GetRequest(), and clDirection3::InitDir().

7.34.2.5 int _stDirReq2::iScaling

Scaling algorithm.

Definition at line 293 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), clDirMsg2::GetRequest(), and clDirection3::ProcessLoop().

7.34.2.6 float _stDirReq2::fScalingExp

Exponent for exponential scaling.

Definition at line 294 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), and clDirMsg2::GetRequest().

7.34.2.7 int _stDirReq2::iRemoveNoise

Background noise estimation and removal algorithm.

Definition at line 295 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), clDirMsg2::GetRequest(), and cl-Direction3::ProcessLoop().

7.34.2.8 float _stDirReq2::fAlpha

Alpha for noise removal.

Definition at line 296 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), clDirMsg2::GetRequest(), and cl-Direction3::ProcessLoop().

7.34.2.9 long _stDirReq2::lMeanLength

Mean length for noise removal.

Definition at line 297 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), clDirMsg2::GetRequest(), and cl-Direction3::ProcessLoop().

7.34.2.10 long _stDirReq2::lGapLength

Gap length for TPSW noise removal algorithm.

Definition at line 298 of file Messages.hh.

Referenced by clSubLocate::ConnectDir(), clDirMsg2::GetRequest(), and cl-Direction3::ProcessLoop().

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

- **Messages.hh**

7.35 _stDirRes Struct Reference

Direction: result.

`#include <Messages.hh>`

Public Attributes

- timeval **sTimeStamp**

  Timestamp.

- float **fIntegrationTime**

  Integration time (s).

- float **fHighFreqLimit**

  Higher frequency limit (Hz).

- float **fPeakLevel**

  Peak level (dB).

- long **lSectorCount**

  Number of sectors (number of results).

- bool **b3DArray**

  3D array (the non-twosided case)

7.35.1 Detailed Description

Direction: result.

Definition at line 272 of file Messages.hh.

7.35.2 Member Data Documentation

7.35.2.1 struct timeval _stDirRes::sTimeStamp

Timestamp.

Definition at line 274 of file Messages.hh.

Referenced by clDirMsg::GetResult(), clDirection2::ProcessLoop(), clDirection::SendResults(), and clDirMsg::SetResult().

7.35.2.2 float _stDirRes::fIntegrationTime

Integration time (s).

Definition at line 275 of file Messages.hh.

Referenced by clDirMsg::GetResult(), clDirection::InitArray(), clGUIDir::OnGdkInput(), clGUIDir::OnSaveClicks(), clDirection2::ProcessLoop(), clGUIDir::SaveInfo(), clDirMsg::SetResult(), and clDirection::StopTiming().

7.35.2.3 float _stDirRes::fHighFreqLimit

Higher frequency limit (Hz).

Definition at line 276 of file Messages.hh.

Referenced by clDirMsg::GetResult(), clDirection2::ProcessLoop(), clGUIDir::SaveInfo(), and clDirMsg::SetResult().

7.35.2.4 float _stDirRes::fPeakLevel

Peak level (dB).

Definition at line 277 of file Messages.hh.

Referenced by clDirMsg::GetResult(), clGUIDir::OnGdkInput(), clGUIDir::PrintStatus(), clDirection::ProcessData(), clDirection2::ProcessLoop(), and clDirMsg::SetResult().

7.35.2.5 long _stDirRes::lSectorCount

Number of sectors (number of results).

Definition at line 278 of file Messages.hh.

Referenced by clDirMsg::GetResult(), clGUIDir::OnGdkInput(), clGUIDir::PrintStatus(), clDirection2::ProcessLoop(), clGUIDir::SaveInfo(), and clDirMsg::SetResult().

7.35.2.6 bool _stDirRes::b3DArray

3D array (the non-twosided case)

Definition at line 279 of file Messages.hh.

Referenced by clDirMsg::GetResult(), clDirection2::ProcessLoop(), and clDirMsg::SetResult().

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

- **Messages.hh**

7.36 _stDirRes2 Struct Reference

Direction3: result.

`#include <Messages.hh>`

Public Attributes

- timeval **sTimeStamp**
  Timestamp.
- long **lMinBin**
  Lowest used bin index.
- long **lMaxBin**
  Highest used bin index.
- float **fFreqResolution**
  Frequency resolution (Hz/bin).
- long **lResultCount**
  Number of results.
- float **fPeakLevel**
  Peak level (dB).
- float **fIntegrationTime**
  Integration time (s).

7.36.1 Detailed Description

Direction3: result.

This is used by locate server.

Definition at line 306 of file Messages.hh.

7.36.2 Member Data Documentation

7.36.2.1 struct timeval _stDirRes2::sTimeStamp

Timestamp.

Definition at line 308 of file Messages.hh.

Referenced by clDirMsg2::GetResult(), clDirection3::ProcessLoop(), and clDirMsg2::SetResult().

7.36.2.2 long _stDirRes2::lMinBin

Lowest used bin index.

Definition at line 309 of file Messages.hh.

Referenced by clDirMsg2::GetResult(), clSubLocate::Main(), clDirection3::ProcessLoop(), and clDirMsg2::SetResult().

7.36.2.3 long _stDirRes2::lMaxBin

Highest used bin index.

Definition at line 310 of file Messages.hh.

Referenced by clDirMsg2::GetResult(), clSubLocate::Main(), clDirection3::ProcessLoop(), and clDirMsg2::SetResult().

7.36.2.4 float _stDirRes2::fFreqResolution

Frequency resolution (Hz/bin).

Definition at line 311 of file Messages.hh.

Referenced by clDirMsg2::GetResult(), clSubLocate::Main(), clDirection3::ProcessLoop(), and clDirMsg2::SetResult().

7.36.2.5 long _stDirRes2::lResultCount

Number of results.

Definition at line 312 of file Messages.hh.

Referenced by clDirMsg2::GetResult(), clSubLocate::Main(), clDirection3::ProcessLoop(), and clDirMsg2::SetResult().

7.36.2.6 float _stDirRes2::fPeakLevel

Peak level (dB).

Definition at line 313 of file Messages.hh.

Referenced by clDirMsg2::GetResult(), clDirection3::ProcessLoop(), and clDirMsg2::SetResult().

7.36.2.7 float _stDirRes2::fIntegrationTime

Integration time (s).

Definition at line 314 of file Messages.hh.

Referenced by clDirMsg2::GetResult(), clDirection3::ProcessLoop(), and clDirMsg2::SetResult().

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

- **Messages.hh**

7.37 estring Class Reference

`#include <Extensions.hh>`

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

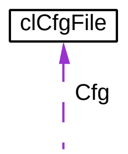
- **Extensions.hh**

7.38 clFileSrv Class Reference

Input server for playback of previously recorded data.

`#include <FileSrv.hh>`

Collaboration diagram for clFileSrv:



Public Member Functions

- **clFileSrv** ()
- ∼**clFileSrv** ()
- int **Main** (int ∗, char ∗∗∗)
- gboolean **OnDelete** (GtkWidget ∗, GdkEvent ∗, gpointer)
- void **OnFileSelectOkClick** (GtkButton ∗, gpointer)
- void **OnFileSelectCancelClick** (GtkButton ∗, gpointer)
- void **OnBrowseClick** (GtkButton ∗, gpointer)
- void **OnPlayStopToggle** (GtkToggleButton ∗, gpointer)
- void **OnPositionChange** (GtkAdjustment ∗, gpointer)
- void ∗ **ReaderThread** (void ∗)
- void ∗ **ServerThread** (void ∗)
- void ∗ **ServeClientThread** (void ∗)

Private Member Functions

- bool **Build** ()
- bool **BuildTable1** ()
- bool **BuildTable2** ()
- bool **ConnectSignals** ()
- double **GetTime** ()
- void **ShortSleep** (long)

Private Attributes

- volatile bool **bRun**
- timeval **sTimeStamp**
- SNDFILE ∗ **sndfileFile**
- SF_INFO **sFileInfo**

- clAlloc **AudioBlock**
- clCondition **CndReady**
- clMutex **MtxAudio**
- clReBufferT< double > **StreamBuf**
- long **lEpoch**
- clAlloc **ConvBuf**
- **clCfgFile Cfg**
- GtkWidget * **gwWindow**
- GtkWidget * **gwVBox**
- GtkWidget * **gwFileSelect**
- GtkWidget * **gwTable1**
- GtkWidget * **gwLFile**
- GtkWidget * **gwEFile**
- GtkWidget * **gwBBrowse**
- GtkObject * **gaPosition**
- GtkWidget * **gwTable2**
- GtkWidget * **gwTBPlayStop**
- GtkWidget * **gwLPosition**
- GtkWidget * **gwHSPosition**

7.38.1 Detailed Description

Input server for playback of previously recorded data.

Definition at line 57 of file FileSrv.hh.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 clFileSrv::clFileSrv ()

Definition at line 349 of file FileSrv.cc.

References bRun, lEpoch, and sndfileFile.

7.38.2.2 clFileSrv::~clFileSrv ()

Definition at line 360 of file FileSrv.cc.

7.38.3 Member Function Documentation

7.38.3.1 bool clFileSrv::Build () `[private]`

Definition at line 130 of file FileSrv.cc.

References BuildTable1(), BuildTable2(), ConnectSignals(), cpFileSelectTxt, cpWindowTxt, FS_WSPACING, gwFileSelect, gwVBox, and gwWindow.

Referenced by Main().

7.38.3.2 bool clFileSrv::BuildTable1 () [private]

Definition at line 154 of file FileSrv.cc.

References cpBrowseTxt, cpFileTxt, FS_WSPACING, gwBBrowse, gwEFile, gwLFile, gwTable1, and gwVBox.

Referenced by Build().

7.38.3.3 bool clFileSrv::BuildTable2 () [private]

Definition at line 193 of file FileSrv.cc.

References cpPlayStopTxt, cpPositionTxt, FS_WSPACING, gaPosition, gwHSPosition, gwLPosition, gwTable2, gwTBPlayStop, and gwVBox.

Referenced by Build().

7.38.3.4 bool clFileSrv::ConnectSignals () [private]

Definition at line 236 of file FileSrv.cc.

References gwBBrowse, gwFileSelect, gwTBPlayStop, gwWindow, WrapOnBrowseClick(), WrapOnDelete(), WrapOnFileSelectCancelClick(), WrapOnFileSelectOkClick(), and WrapOnPlayStopToggle().

Referenced by Build().

7.38.3.5 double clFileSrv::GetTime () [private]

Definition at line 263 of file FileSrv.cc.

References lEpoch.

Referenced by ReaderThread().

7.38.3.6 void clFileSrv::ShortSleep (long) [private]

Definition at line 283 of file FileSrv.cc.

Referenced by ReaderThread().

7.38.3.7 int clFileSrv::Main (int *, char ***)

Definition at line 365 of file FileSrv.cc.

References AudioBlock, Build(), Cfg, cpWindowTxt, FileSrvThreads(), FS_CFGFILE, FS_DEFAULT_PORT, FS_FRAG_SIZE_DEFAULT, FS_VER_MAJ, FS_VER_MIN, FS_VER_PL, GDT, clCfgFile::GetInt(), ReaderThread(), ServerThread(), and clCfgFile::SetFileName().

Referenced by main().

7.38.3.8 gboolean clFileSrv::OnDelete (GtkWidget *, GdkEvent *, gpointer)

Definition at line 406 of file FileSrv.cc.

References bRun.

Referenced by WrapOnDelete().

7.38.3.9 void clFileSrv::OnFileSelectOkClick (GtkButton *, gpointer)

Definition at line 415 of file FileSrv.cc.

References gwEFile, and gwFileSelect.

Referenced by WrapOnFileSelectOkClick().

7.38.3.10 void clFileSrv::OnFileSelectCancelClick (GtkButton *, gpointer)

Definition at line 432 of file FileSrv.cc.

References gwFileSelect.

Referenced by WrapOnFileSelectCancelClick().

7.38.3.11 void clFileSrv::OnBrowseClick (GtkButton *, gpointer)

Definition at line 438 of file FileSrv.cc.

References gwFileSelect.

Referenced by WrapOnBrowseClick().

7.38.3.12 void clFileSrv::OnPlayStopToggle (GtkToggleButton *, gpointer)

Definition at line 444 of file FileSrv.cc.

References gaPosition, gwEFile, MtxAudio, sFileInfo, and sndfileFile.

Referenced by WrapOnPlayStopToggle().

7.38.3.13 void clFileSrv::OnPositionChange (GtkAdjustment *, gpointer)

Definition at line 504 of file FileSrv.cc.

Referenced by WrapOnPositionChange().

7.38.3.14 void * clFileSrv::ReaderThread (void *)

Definition at line 509 of file FileSrv.cc.

References AudioBlock, bRun, CndReady, FileSrvThreads(), FS_INTHREAD_-PRIORITY, gaPosition, GDT, GetTime(), MtxAudio, sFileInfo, ShortSleep(), sndfileFile, sTimeStamp, and StreamBuf.

Referenced by Main().

7.38.3.15 void * clFileSrv::ServerThread (void *)

Definition at line 608 of file FileSrv.cc.

References clSockServ::Bind(), bRun, FileSrvThreads(), FS_ACCEPT_-TIMEOUT, ServeClientThread(), and clSockServ::WaitForConnect().

Referenced by Main().

7.38.3.16 void * clFileSrv::ServeClientThread (void *)

Definition at line 632 of file FileSrv.cc.

References AudioBlock, bRun, CndReady, clSockOp::DisableNagle(), File-SrvThreads(), FS_OUTTHREAD_PRIORITY, GDT, GLOBAL_HEADER_-LEN, MSG_SOUND_COMPRESS_NONE, MtxAudio, clSoundMsg::SetData(), clSoundMsg::SetStart(), clSockOp::SetTypeOfService(), sFileInfo, SOp, sTime-Stamp, and clSockOp::WriteN().

Referenced by ServerThread().

7.38.4 Member Data Documentation

7.38.4.1 volatile bool clFileSrv::bRun [private]

Definition at line 60 of file FileSrv.hh.

Referenced by clFileSrv(), OnDelete(), ReaderThread(), ServeClientThread(), and ServerThread().

7.38.4.2 struct timeval clFileSrv::sTimeStamp [private]

Definition at line 61 of file FileSrv.hh.

Referenced by ReaderThread(), and ServeClientThread().

7.38.4.3 SNDFILE* clFileSrv::sndfileFile [private]

Definition at line 62 of file FileSrv.hh.

Referenced by clFileSrv(), OnPlayStopToggle(), and ReaderThread().

7.38.4.4 SF_INFO clFileSrv::sFileInfo [private]

Definition at line 63 of file FileSrv.hh.

Referenced by OnPlayStopToggle(), ReaderThread(), and ServeClientThread().

7.38.4.5 clAlloc clFileSrv::AudioBlock [private]

Definition at line 67 of file FileSrv.hh.

Referenced by Main(), ReaderThread(), and ServeClientThread().

7.38.4.6 clCondition clFileSrv::CndReady [private]

Definition at line 68 of file FileSrv.hh.

Referenced by ReaderThread(), and ServeClientThread().

7.38.4.7 clMutex clFileSrv::MtxAudio [private]

Definition at line 69 of file FileSrv.hh.

Referenced by OnPlayStopToggle(), ReaderThread(), and ServeClientThread().

7.38.4.8 clReBufferT<double> clFileSrv::StreamBuf [private]

Definition at line 70 of file FileSrv.hh.

Referenced by ReaderThread().

7.38.4.9 long clFileSrv::lEpoch [private]

Definition at line 72 of file FileSrv.hh.

Referenced by clFileSrv(), and GetTime().

7.38.4.10 clAlloc clFileSrv::ConvBuf [private]

Definition at line 73 of file FileSrv.hh.

7.38.4.11 clCfgFile clFileSrv::Cfg [private]

Definition at line 74 of file FileSrv.hh.

Referenced by Main().

7.38.4.12 GtkWidget* clFileSrv::gwWindow [private]

Definition at line 76 of file FileSrv.hh.

Referenced by Build(), and ConnectSignals().

7.38.4.13 GtkWidget* clFileSrv::gwVBox [private]

Definition at line 77 of file FileSrv.hh.

Referenced by Build(), BuildTable1(), and BuildTable2().

7.38.4.14 GtkWidget* clFileSrv::gwFileSelect [private]

Definition at line 78 of file FileSrv.hh.

Referenced by Build(), ConnectSignals(), OnBrowseClick(), OnFileSelectCancelClick(), and OnFileSelectOkClick().

7.38.4.15 GtkWidget* clFileSrv::gwTable1 [private]

Definition at line 80 of file FileSrv.hh.

Referenced by BuildTable1().

7.38.4.16 GtkWidget* clFileSrv::gwLFile [private]

Definition at line 81 of file FileSrv.hh.

Referenced by BuildTable1().

7.38.4.17 GtkWidget* clFileSrv::gwEFile [private]

Definition at line 82 of file FileSrv.hh.

Referenced by BuildTable1(), OnFileSelectOkClick(), and OnPlayStopToggle().

7.38.4.18 GtkWidget* clFileSrv::gwBBrowse [private]

Definition at line 83 of file FileSrv.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.38.4.19 GtkObject* clFileSrv::gaPosition [private]

Definition at line 85 of file FileSrv.hh.

Referenced by BuildTable2(), OnPlayStopToggle(), and ReaderThread().

7.38.4.20 GtkWidget* clFileSrv::gwTable2 [private]

Definition at line 86 of file FileSrv.hh.

Referenced by BuildTable2().

7.38.4.21 GtkWidget* clFileSrv::gwTBPlayStop [private]

Definition at line 87 of file FileSrv.hh.

Referenced by BuildTable2(), and ConnectSignals().

7.38.4.22 GtkWidget* clFileSrv::gwLPosition [private]

Definition at line 88 of file FileSrv.hh.

Referenced by BuildTable2().

7.38.4.23 GtkWidget* clFileSrv::gwHSPosition [private]

Definition at line 89 of file FileSrv.hh.

Referenced by BuildTable2().

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

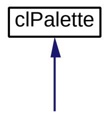
- **FileSrv.hh**
- **FileSrv.cc**

7.39 clFrameBuf Class Reference

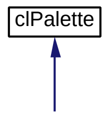
Framebuffer class.

`#include <FrameBuf.hh>`

Inheritance diagram for clFrameBuf:



Collaboration diagram for clFrameBuf:



Public Member Functions

- **clFrameBuf** ()
- ~**clFrameBuf** ()
- void **SetSize** (int, int)

  Set size of framebuffer.

- void **Clear** ()

  Clear framebuffer.

- unsigned int * **GetFBPtr** ()

  Get pointer to raw framebuffer.

- unsigned int * **GetPalPtr** ()

  Get pointer to palette buffer.

- guchar * **GetCurPtr** (int, int)

  Get pointer to current position in doublebuffered framebuffer.

- gint **GetRowStride** ()

  Get row stride.

- int **GetNumColors** ()

Get number of colors in palette LUT.

- int **GetWidth** ()

  Get width of framebuffer.

- int **GetHeight** ()

  Get height of framebuffer.

- void **DrawLine** (unsigned int ∗)

  Draw line to vertically scrolling graph.

- void **DrawLine** (GDT ∗)
- void **DrawColumn** (unsigned int ∗)

  Draw column to horizontally scrolling graph.

- void **DrawColumn** (GDT ∗)
- unsigned int **operator**[ ] (int iPix)

  Get single pixel from framebuffer.

- void **PalGenBW** ()

  Generate white-to-black palette.

- void **PalGenHSV** ()

  Generate black-blue-cyan-green-yellow-red palette.

- void **PalGenLight** ()

  Generate palette matching to light's spectrum.

- void **PalGenTemp** ()

  Generate black-red-yellow-white palette.

- void **PalGenDir** ()

  Generate black-white palette with two highest values red.

- void **PalGenGreen** ()

  Generate black-green-red palette.

- void **PalGenGreen2** ()

  Generate black-green-white palette.

- void **PalGenGreen3** ()

  Generate black-green-yellow palette.

- void **PalGenGreen4** ()

  Generate black-green-red-yellow palette.

- void **PalGenPureGreen** ()

  Generate black-green palette.

- void **PalGenWB** ()

  Generate black-white palette.

- bool **SaveToFile** (const char *, int, int, const char *)

  Save current framebuffer contents to TIFF file.

- bool **SaveToFile** (const char *, int, int, const char *, double, double, double, double)
- bool **StartSaveToFile** (const char *, int, int, const char *, int)

  Start continuous save to TIFF file.

- void **StopSaveToFile** ()

  Stop continuous save to TIFF file.

Private Member Functions

- void **RGBA2RGB** (unsigned char *, const unsigned int *, int, int, int, int, int, int)
- bool **SetTiffTags** (const char *, int, int, const char *, int, int, int)
- bool **WriteTiffData** ()
- bool **WriteTiffData** (int, int, double, double, double, double)
- bool **WriteTiffScanData** ()
- void **DrawHLine** (unsigned char *, int, int, int, int, int)
- void **DrawVLine** (unsigned char *, int, int, int, int, int)
- bool **DrawText** (unsigned char *, int, int, int, int, const char *)

Private Attributes

- bool **bSaving**
- int **iType**
- int **iWidth**
- int **iHeight**
- int **iCurPos**
- int **iPixFontSize**
- int **iMaxFontWidth**
- char **cpDateTime** [FB_TIFF_DATELEN+1]
- unsigned int * **upFrameBuf**
- uint32 * **upScanBuf**
- uint32 **uiStripSize**
- tstrip_t **uiContStrip**
- TIFF * **tiffImg**
- FT_Library **ftLib**

- FT_Face **ftFace**
- clAlloc **FrameBuf**
- clAlloc **ScanBuf**

7.39.1 Detailed Description

Framebuffer class.

Framebuffer class for vertically or horizontally scrolling intensity graph. Capable of saving contents to tiff and also continuous saving. Uses doublebuffering for speed.

Dataformat is 32-bit BGRA (0xAABBGGRR).

Definition at line 87 of file FrameBuf.hh.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 clFrameBuf::clFrameBuf ()

Definition at line 393 of file FrameBuf.cc.

References bSaving, FB_TIFF_FONTSIZE, FB_TIFF_RESOLUTION, FB_TYPE_NONE, ftFace, ftLib, iCurPos, iHeight, iMaxFontWidth, iPixFontSize, iType, iWidth, upFrameBuf, and upScanBuf.

7.39.2.2 clFrameBuf::~clFrameBuf ()

Definition at line 430 of file FrameBuf.cc.

References bSaving, ftFace, ftLib, and StopSaveToFile().

7.39.3 Member Function Documentation

7.39.3.1 void clFrameBuf::RGBA2RGB (unsigned char *, const unsigned int *, int, int, int, int, int, int) `[private]`

Definition at line 39 of file FrameBuf.cc.

Referenced by WriteTiffData(), and WriteTiffScanData().

7.39.3.2 bool clFrameBuf::SetTiffTags (const char *, int, int, const char *, int, int, int) `[private]`

Definition at line 65 of file FrameBuf.cc.

References cpDateTime, FB_TIFF_COMPRESS_JPEG, FB_TIFF_CONT_HORIZONTAL, FB_TIFF_CONT_NO, FB_TIFF_CONT_VERTICAL, FB_TIFF_DATELEN, FB_TIFF_RESOLUTION, GLOBAL_VERSMAJ, GLOBAL_VERSMIN, GLOBAL_VERSPL, tiffImg, and uiStripSize.

Referenced by SaveToFile(), and StartSaveToFile().

7.39.3.3 bool clFrameBuf::WriteTiffData () [private]

Definition at line 165 of file FrameBuf.cc.

References GetCurPtr(), iHeight, iWidth, RGBA2RGB(), tiffImg, and uiStripSize.

Referenced by SaveToFile().

7.39.3.4 bool clFrameBuf::WriteTiffData (int, int, double, double, double, double) [private]

Definition at line 206 of file FrameBuf.cc.

References cpDateTime, DrawHLine(), DrawText(), DrawVLine(), GetCurPtr(), iHeight, iMaxFontWidth, iPixFontSize, iWidth, RGBA2RGB(), tiffImg, and uiStripSize.

7.39.3.5 bool clFrameBuf::WriteTiffScanData () [private]

Definition at line 287 of file FrameBuf.cc.

References iWidth, RGBA2RGB(), tiffImg, uiContStrip, and upScanBuf.

Referenced by DrawColumn(), and DrawLine().

7.39.3.6 void clFrameBuf::DrawHLine (unsigned char *, int, int, int, int, int) [private]

Definition at line 308 of file FrameBuf.cc.

Referenced by WriteTiffData().

7.39.3.7 void clFrameBuf::DrawVLine (unsigned char *, int, int, int, int, int) [private]

Definition at line 324 of file FrameBuf.cc.

Referenced by WriteTiffData().

7.39.3.8 bool clFrameBuf::DrawText (unsigned char *, int, int, int, int, const char *) [private]

Definition at line 341 of file FrameBuf.cc.

References FB_TIFF_RESOLUTION, ftFace, and iPixFontSize.

Referenced by WriteTiffData().

7.39.3.9 void clFrameBuf::SetSize (int, int)

Set size of framebuffer.

Resize is non-destructive.

Parameters:

iReqWidth Width

iReqHeight Height

Definition at line 438 of file FrameBuf.cc.

References FB_TYPE_COLUMN, FB_TYPE_LINE, FB_TYPE_NONE, FrameBuf, iCurPos, iHeight, iType, iWidth, ScanBuf, upFrameBuf, and upScanBuf.

Referenced by clSpectGUI::Build(), clGUILofar::Configure(), clGUILofar::Exec(), clGUIDir::Exec(), clGUIDir::InitConnection(), and clSpectGUI::ReConfigDisplay().

7.39.3.10 void clFrameBuf::Clear ()

Clear framebuffer.

Definition at line 467 of file FrameBuf.cc.

References iHeight, iWidth, and upFrameBuf.

Referenced by clGUILofar::Exec(), and clGUIDir::Exec().

7.39.3.11 unsigned int* clFrameBuf::GetFBPtr () `[inline]`

Get pointer to raw framebuffer.

Returns:

Pointer to raw framebuffer

Definition at line 137 of file FrameBuf.hh.

References upFrameBuf.

Referenced by clSpectGUI::Build().

7.39.3.12 unsigned int* clFrameBuf::GetPalPtr () `[inline]`

Get pointer to palette buffer.

Returns:

Pointer to palette LUT

Definition at line 143 of file FrameBuf.hh.

References clPalette::upPalette.

Referenced by clSpectGUI::Build().

7.39.3.13 guchar * clFrameBuf::GetCurPtr (int, int)

Get pointer to current position in doublebuffered framebuffer.

Parameters:

iXPos X position offset

iYPos Y position offset

Returns:

Pointer to framebuffer position

Definition at line 480 of file FrameBuf.cc.

References FB_TYPE_COLUMN, FB_TYPE_LINE, iCurPos, iType, iWidth, and upFrameBuf.

Referenced by clSpectGUI::DrawSpectrogram(), clGUIDir::InitConnection(), clGUILofar::OnExposeLofar(), clSpectGUI::OnExposeSgram(), clGUIDir::OnExposeWorm(), clGUILofar::OnGdkInput(), clGUIDir::OnGdkInput(), and WriteTiffData().

7.39.3.14 gint clFrameBuf::GetRowStride () `[inline]`

Get row stride.

Get size of scanline in bytes. For use with gdk_rgb_*

Returns:

Size of scanline

Definition at line 159 of file FrameBuf.hh.

References iWidth.

Referenced by clSpectGUI::DrawSpectrogram(), clGUIDir::InitConnection(), clGUILofar::OnExposeLofar(), clSpectGUI::OnExposeSgram(), clGUIDir::OnExposeWorm(), clGUILofar::OnGdkInput(), and clGUIDir::OnGdkInput().

7.39.3.15 int clFrameBuf::GetNumColors () `[inline]`

Get number of colors in palette LUT.

Returns:

Number of colors

Definition at line 165 of file FrameBuf.hh.

References clPalette::iPalSize.

Referenced by clSpectGUI::Build().

7.39.3.16 int clFrameBuf::GetWidth () [inline]

Get width of framebuffer.

Returns:

Width of framebuffer

Definition at line 171 of file FrameBuf.hh.

References iWidth.

7.39.3.17 int clFrameBuf::GetHeight () [inline]

Get height of framebuffer.

Returns:

Height of framebuffer

Definition at line 177 of file FrameBuf.hh.

References iHeight.

Referenced by clGUILofar::OnSaveClicks(), and clGUIDir::OnSaveClicks().

7.39.3.18 void clFrameBuf::DrawLine (unsigned int *)

Draw line to vertically scrolling graph.

Scroll graph down and draw new scanline to top. Indexes to palette LUT as parameter.

Parameters:

upLineBuf Scanline LUT indexes

Definition at line 498 of file FrameBuf.cc.

References bSaving, FB_TYPE_LINE, iCurPos, iHeight, iType, iWidth, upFrameBuf, clPalette::upPalette, upScanBuf, and WriteTiffScanData().

Referenced by clGUILofar::OnGdkInput(), and clGUIDir::OnGdkInput().

7.39.3.19 void clFrameBuf::DrawLine (GDT *)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters:

fpDataBuf Normalized data values [0..1]

Definition at line 524 of file FrameBuf.cc.

References bSaving, FB_TYPE_LINE, GDT, iCurPos, iHeight, clPalette::iPalSize, iType, iWidth, upFrameBuf, clPalette::upPalette, upScanBuf, and WriteTiffScanData().

7.39.3.20 void clFrameBuf::DrawColumn (unsigned int ∗)

Draw column to horizontally scrolling graph.

Scroll graph left and draw new column to right. Indexes to palette LUT as parameter.

Parameters:

upColumnBuf Column LUT indexes

Definition at line 555 of file FrameBuf.cc.

References bSaving, FB_TYPE_COLUMN, iCurPos, iHeight, iType, iWidth, upFrameBuf, clPalette::upPalette, upScanBuf, and WriteTiffScanData().

Referenced by clSpectGUI::DrawSpectrogram().

7.39.3.21 void clFrameBuf::DrawColumn (GDT ∗)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters:

fpDataBuf Normalized data values [0..1]

Definition at line 581 of file FrameBuf.cc.

References bSaving, FB_TYPE_COLUMN, GDT, iCurPos, iHeight, clPalette::iPalSize, iType, iWidth, upFrameBuf, clPalette::upPalette, upScanBuf, and WriteTiffScanData().

7.39.3.22 unsigned int clFrameBuf::operator[] (int *iPix*) `[inline]`

Get single pixel from framebuffer.

Returns:

Pixel value

Reimplemented from **clPalette** (p. 416).

Definition at line 211 of file FrameBuf.hh.

References upFrameBuf.

7.39.3.23 void clFrameBuf::PalGenBW () [inline]

Generate white-to-black palette.

Definition at line 216 of file FrameBuf.hh.

References clPalette::GenBW().

Referenced by clSpectGUI::BuildDrawingPrims(), clGUIDir::BuildDrawingPrims(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), and clGUILofar::SetPalette().

7.39.3.24 void clFrameBuf::PalGenHSV () [inline]

Generate black-blue-cyan-green-yellow-red palette.

Definition at line 220 of file FrameBuf.hh.

References clPalette::GenHSV().

Referenced by clSpectGUI::BuildDrawingPrims(), clGUIDir::BuildDrawingPrims(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), and clGUILofar::SetPalette().

7.39.3.25 void clFrameBuf::PalGenLight () [inline]

Generate palette matching to light's spectrum.

Definition at line 224 of file FrameBuf.hh.

References clPalette::GenLight().

Referenced by clSpectGUI::BuildDrawingPrims(), clGUIDir::BuildDrawingPrims(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), and clGUILofar::SetPalette().

7.39.3.26 void clFrameBuf::PalGenTemp () [inline]

Generate black-red-yellow-white palette.

Definition at line 228 of file FrameBuf.hh.

References clPalette::GenTemp().

Referenced by clSpectGUI::BuildDrawingPrims(), clGUIDir::BuildDrawingPrims(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), and clGUILofar::SetPalette().

7.39.3.27 void clFrameBuf::PalGenDir () [inline]

Generate black-white palette with two highest values red.

Definition at line 232 of file FrameBuf.hh.

References clPalette::GenDir().

Referenced by clSpectGUI::BuildDrawingPrims(), clGUIDir::BuildDrawingPrims(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), and clGUILofar::SetPalette().

7.39.3.28 void clFrameBuf::PalGenGreen () [inline]

Generate black-green-red palette.

"NATO-style"

Definition at line 236 of file FrameBuf.hh.

References clPalette::GenGreen().

Referenced by clSpectGUI::BuildDrawingPrims(), clGUIDir::BuildDrawingPrims(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), and clGUILofar::SetPalette().

7.39.3.29 void clFrameBuf::PalGenGreen2 () [inline]

Generate black-green-white palette.

Definition at line 240 of file FrameBuf.hh.

References clPalette::GenGreen2().

Referenced by clGUIDir::BuildDrawingPrims(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), and clGUILofar::SetPalette().

7.39.3.30 void clFrameBuf::PalGenGreen3 () [inline]

Generate black-green-yellow palette.

Definition at line 244 of file FrameBuf.hh.

References clPalette::GenGreen3().

Referenced by clGUILofar::SetPalette().

7.39.3.31 void clFrameBuf::PalGenGreen4 () [inline]

Generate black-green-red-yellow palette.

Definition at line 248 of file FrameBuf.hh.

References clPalette::GenGreen4().

Referenced by clGUILofar::SetPalette().

7.39.3.32 void clFrameBuf::PalGenPureGreen () [inline]

Generate black-green palette.

Definition at line 252 of file FrameBuf.hh.

References clPalette::GenPureGreen().

Referenced by clGUIDir::BuildDrawingPrims(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), and clGUILofar::SetPalette().

7.39.3.33 void clFrameBuf::PalGenWB () [inline]

Generate black-white palette.

Definition at line 256 of file FrameBuf.hh.

References clPalette::GenWB().

Referenced by clGUIDir::BuildDrawingPrims(), clSpectGUI::OnPaletteActivate(), clGUIDir::OnPaletteActivate(), and clGUILofar::SetPalette().

7.39.3.34 bool clFrameBuf::SaveToFile (const char *, int, int, const char *)

Save current framebuffer contents to TIFF file.

Parameters:

cpFileName Name of TIFF file

iCompression TIFF CODEC

iJPEGQuality Quality for JPEG CODEC

cpDescription Description

dXMin X-axis minimum

dXMax X-axis maximum

dYMin Y-axis minimum

dYMax Y-axis maximum

Returns:

Success

Definition at line 612 of file FrameBuf.cc.

References bSaving, FB_TIFF_CONT_NO, FB_TIFF_MODE, iHeight, iWidth, SetTiffTags(), tiffImg, and WriteTiffData().

Referenced by clSpectGUI::OnSaveClicks(), clGUILofar::OnSaveClicks(), and clGUIDir::OnSaveClicks().

7.39.3.35 bool clFrameBuf::SaveToFile (const char *, int, int, const char *, double, double, double, double)

Definition at line 632 of file FrameBuf.cc.

References bSaving, FB_TIFF_CONT_NO, FB_TIFF_MODE, iHeight, iMaxFontWidth, iPixFontSize, iWidth, SetTiffTags(), tiffImg, and WriteTiffData().

7.39.3.36 bool clFrameBuf::StartSaveToFile (const char *, int, int, const char *, int)

Start continuous save to TIFF file.

All new data is saved to TIFF until saving is stopped.

Parameters:

cpFileName Name of TIFF file

iCompression TIFF CODEC

iJPEGQuality Quality for JPEG CODEC

cpDescription Description

iDirection Drawing direction

Returns:

Success

Definition at line 660 of file FrameBuf.cc.

References bSaving, FB_TIFF_MODE, iHeight, iWidth, SetTiffTags(), tiffImg, and uiContStrip.

Referenced by clGUILofar::OnSaveClicks(), clGUIDir::OnSaveClicks(), clGUILofar::StartNewImgFile(), and clGUIDir::StartNewImgFile().

7.39.3.37 void clFrameBuf::StopSaveToFile ()

Stop continuous save to TIFF file.

Definition at line 676 of file FrameBuf.cc.

References bSaving, and tiffImg.

Referenced by clGUILofar::OnDelete(), clGUIDir::OnDelete(), clGUILofar::OnSaveClicks(), clGUIDir::OnSaveClicks(), clGUILofar::StartNewImgFile(), clGUIDir::StartNewImgFile(), and ~clFrameBuf().

7.39.4 Member Data Documentation

7.39.4.1 bool clFrameBuf::bSaving `[private]`

Definition at line 89 of file FrameBuf.hh.

Referenced by clFrameBuf(), DrawColumn(), DrawLine(), SaveToFile(), StartSaveToFile(), StopSaveToFile(), and ~clFrameBuf().

7.39.4.2 int clFrameBuf::iType [private]

Definition at line 90 of file FrameBuf.hh.

Referenced by clFrameBuf(), DrawColumn(), DrawLine(), GetCurPtr(), and SetSize().

7.39.4.3 int clFrameBuf::iWidth [private]

Definition at line 91 of file FrameBuf.hh.

Referenced by Clear(), clFrameBuf(), DrawColumn(), DrawLine(), GetCurPtr(), GetRowStride(), GetWidth(), SaveToFile(), SetSize(), StartSaveToFile(), WriteTiffData(), and WriteTiffScanData().

7.39.4.4 int clFrameBuf::iHeight [private]

Definition at line 92 of file FrameBuf.hh.

Referenced by Clear(), clFrameBuf(), DrawColumn(), DrawLine(), GetHeight(), SaveToFile(), SetSize(), StartSaveToFile(), and WriteTiffData().

7.39.4.5 int clFrameBuf::iCurPos [private]

Definition at line 93 of file FrameBuf.hh.

Referenced by clFrameBuf(), DrawColumn(), DrawLine(), GetCurPtr(), and SetSize().

7.39.4.6 int clFrameBuf::iPixFontSize [private]

Definition at line 94 of file FrameBuf.hh.

Referenced by clFrameBuf(), DrawText(), SaveToFile(), and WriteTiffData().

7.39.4.7 int clFrameBuf::iMaxFontWidth [private]

Definition at line 95 of file FrameBuf.hh.

Referenced by clFrameBuf(), SaveToFile(), and WriteTiffData().

7.39.4.8 char clFrameBuf::cpDateTime[FB_TIFF_DATELEN + 1] [private]

Definition at line 96 of file FrameBuf.hh.

Referenced by SetTiffTags(), and WriteTiffData().

7.39.4.9 unsigned int* clFrameBuf::upFrameBuf [private]

Definition at line 97 of file FrameBuf.hh.

Referenced by Clear(), clFrameBuf(), DrawColumn(), DrawLine(), GetCurPtr(), GetFBPtr(), operator[](), and SetSize().

7.39.4.10 uint32* clFrameBuf::upScanBuf [private]

Definition at line 98 of file FrameBuf.hh.

Referenced by clFrameBuf(), DrawColumn(), DrawLine(), SetSize(), and WriteTiffScanData().

7.39.4.11 uint32 clFrameBuf::uiStripSize [private]

Definition at line 99 of file FrameBuf.hh.

Referenced by SetTiffTags(), and WriteTiffData().

7.39.4.12 tstrip_t clFrameBuf::uiContStrip [private]

Definition at line 100 of file FrameBuf.hh.

Referenced by StartSaveToFile(), and WriteTiffScanData().

7.39.4.13 TIFF* clFrameBuf::tiffImg [private]

Definition at line 101 of file FrameBuf.hh.

Referenced by SaveToFile(), SetTiffTags(), StartSaveToFile(), StopSaveToFile(), WriteTiffData(), and WriteTiffScanData().

7.39.4.14 FT_Library clFrameBuf::ftLib [private]

Definition at line 102 of file FrameBuf.hh.

Referenced by clFrameBuf(), and ~clFrameBuf().

7.39.4.15 FT_Face clFrameBuf::ftFace [private]

Definition at line 103 of file FrameBuf.hh.

Referenced by clFrameBuf(), DrawText(), and ~clFrameBuf().

7.39.4.16 clAlloc clFrameBuf::FrameBuf [private]

Definition at line 104 of file FrameBuf.hh.

Referenced by SetSize().

7.39.4.17 clAlloc clFrameBuf::ScanBuf [private]

Definition at line 105 of file FrameBuf.hh.

Referenced by SetSize().

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

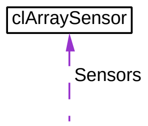
- **FrameBuf.hh**
- **FrameBuf.cc**

7.40 clFreqBeamDipole Class Reference

Frequency-domain beamforming for dipole array.

`#include <FreqBeamDipole.hh>`

Collaboration diagram for clFreqBeamDipole:



Public Member Functions

- **clFreqBeamDipole** ()
- ~**clFreqBeamDipole** ()
- void **SetDebug** (bool)

  Enable/disable debug printouts.

- bool **Initialize** (GDT, long, GDT)

  Initialize dipole array.

- void **SetSoundSpeed** (GDT)

  Set speed of sound.

- void **SetDirection** (GDT, bool)

  Set direction.

- void **Put** (const GDT *, long, long, long)

  Put data into beamformer.

- bool **Get** (GDT *, long)

  Get data from beamformer.

Protected Attributes

- GDT **fSensorSpacing**
- **clArraySensor Sensors** [2]

7.40.1 Detailed Description

Frequency-domain beamforming for dipole array.

Definition at line 34 of file FreqBeamDipole.hh.

7.40.2 Constructor & Destructor Documentation

7.40.2.1 clFreqBeamDipole::clFreqBeamDipole () `[inline]`

Definition at line 40 of file FreqBeamDipole.hh.

7.40.2.2 clFreqBeamDipole::~clFreqBeamDipole () `[inline]`

Definition at line 41 of file FreqBeamDipole.hh.

7.40.3 Member Function Documentation

7.40.3.1 void clFreqBeamDipole::SetDebug (bool)

Enable/disable debug printouts.

Parameters:
  bDebug Enable/disable debug printouts

Definition at line 31 of file FreqBeamDipole.cc.

References clArraySensor::DisableDebug(), clArraySensor::EnableDebug(), and Sensors.

Referenced by clBeamAudio::InitBeam().

7.40.3.2 bool clFreqBeamDipole::Initialize (GDT, long, GDT)

Initialize dipole array.

Parameters:
  fSpacing Sensor spacing (m)
  lWinSize Window size
  fSampleRate Sampling rate

Returns:
  Success

Definition at line 43 of file FreqBeamDipole.cc.

References fSensorSpacing, clArraySensor::Initialize(), Sensors, and clArraySensor::SetSampleRate().

Referenced by clBeamAudio::InitBeam(), clBeamSrv2Slave::Main(), and clBeamSrvSlave::Main().

7.40.3.3 void clFreqBeamDipole::SetSoundSpeed (GDT)

Set speed of sound.

Parameters:

fSoundSpeed Speed of sound (m/s)

Definition at line 60 of file FreqBeamDipole.cc.

References fSensorSpacing, Sensors, clArraySensor::SetArrayFrequency(), and clArraySensor::SetSoundSpeed().

Referenced by clBeamAudio::InitBeam(), clBeamSrv2Slave::Main(), and clBeamSrvSlave::Main().

7.40.3.4 void clFreqBeamDipole::SetDirection (GDT, bool)

Set direction.

Parameters:

fDirection Direction (rad)

bLowPass Lowpass filter

Definition at line 73 of file FreqBeamDipole.cc.

References Sensors, and clArraySensor::SetDirection().

Referenced by clBeamAudio::InitBeam(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), and clBeamAudio::SetDirection().

7.40.3.5 void clFreqBeamDipole::Put (const GDT *, long, long, long)

Put data into beamformer.

Parameters:

fpSrc Source vector

lSrcCount Number of samples in source vector

lSensorOffs Sensor offset in source vector

lChannels Number of channels in source vector

Definition at line 89 of file FreqBeamDipole.cc.

References clArraySensor::Put(), and Sensors.

Referenced by clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), and clBeamAudio::ProcessLoop().

7.40.3.6 bool clFreqBeamDipole::Get (GDT *, long)

Get data from beamformer.

Parameters:

fpDest Destination vector

lDestCount Number of samples to put into destination vector

Returns:

Success

Definition at line 105 of file FreqBeamDipole.cc.

References GDT, clArraySensor::Get(), and Sensors.

Referenced by clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), and clBeamAudio::ProcessLoop().

7.40.4 Member Data Documentation

7.40.4.1 GDT clFreqBeamDipole::fSensorSpacing `[protected]`

Definition at line 37 of file FreqBeamDipole.hh.

Referenced by Initialize(), and SetSoundSpeed().

7.40.4.2 clArraySensor clFreqBeamDipole::Sensors[2] `[protected]`

Definition at line 38 of file FreqBeamDipole.hh.

Referenced by Get(), Initialize(), Put(), SetDebug(), SetDirection(), and SetSoundSpeed().

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

- **FreqBeamDipole.hh**
- **FreqBeamDipole.cc**

7.41 clFreqBeamLine Class Reference

Frequency-domain beamforming for line array.

`#include <FreqBeamLine.hh>`

Public Member Functions

- **clFreqBeamLine** ()
- ~**clFreqBeamLine** ()
- void **SetDebug** (bool)
  Enable/disable debug printouts.
- bool **Initialize** (long, GDT, long, GDT)
  Initialize dipole array.
- void **SetSoundSpeed** (GDT)
  Set speed of sound.
- void **SetDirection** (GDT, bool)
  Set direction.
- void **Put** (const GDT *, long, long, long)
  Put data into beamformer.
- bool **Get** (GDT *, long)
  Get data from beamformer.

Protected Attributes

- long **lSensorCount**
- GDT **fSensorSpacing**
- std::vector< **clArraySensor** * > **vSensors**

7.41.1 Detailed Description

Frequency-domain beamforming for line array.

Array is shaded using Dolph-Chebyshev window.

Definition at line 38 of file FreqBeamLine.hh.

7.41.2 Constructor & Destructor Documentation

7.41.2.1 clFreqBeamLine::clFreqBeamLine () `[inline]`

Definition at line 45 of file FreqBeamLine.hh.

7.41.2.2 clFreqBeamLine::~clFreqBeamLine ()

Definition at line 32 of file FreqBeamLine.cc.

References vSensors.

7.41.3 Member Function Documentation

7.41.3.1 void clFreqBeamLine::SetDebug (bool)

Enable/disable debug printouts.

Parameters:

bDebug Enable/disable debug printouts

Definition at line 42 of file FreqBeamLine.cc.

References vSensors.

Referenced by clBeamAudio::InitBeam().

7.41.3.2 bool clFreqBeamLine::Initialize (long, GDT, long, GDT)

Initialize dipole array.

Parameters:

lSensors Number of sensors in array

fSpacing Sensor spacing (m)

lWinSize Window size

fSampleRate Sampling rate

Returns:

Success

Definition at line 54 of file FreqBeamLine.cc.

References fSensorSpacing, GDT, clArraySensor::Initialize(), lSensorCount, clArraySensor::SetSampleRate(), clArraySensor::SetShading(), and vSensors.

Referenced by clBeamAudio::InitBeam(), clBeamSrv2Slave::Main(), and clBeamSrvSlave::Main().

7.41.3.3 void clFreqBeamLine::SetSoundSpeed (GDT)

Set speed of sound.

Parameters:

fSoundSpeed Speed of sound (m/s)

Definition at line 80 of file FreqBeamLine.cc.

References fSensorSpacing, and vSensors.

Referenced by clBeamAudio::InitBeam(), clBeamSrv2Slave::Main(), and clBeamSrvSlave::Main().

7.41.3.4 void clFreqBeamLine::SetDirection (GDT, bool)

Set direction.

Parameters:

fDirection Direction (rad)

bLowPass Lowpass filter

Definition at line 93 of file FreqBeamLine.cc.

References vSensors.

Referenced by clBeamAudio::InitBeam(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), and clBeamAudio::SetDirection().

7.41.3.5 void clFreqBeamLine::Put (const GDT *, long, long, long)

Put data into beamformer.

Parameters:

fpSrc Source vector

lSrcCount Number of samples in source vector

lSensorOffs Sensor offset in source vector

lChannels Number of channels in source vector

Definition at line 109 of file FreqBeamLine.cc.

References vSensors.

Referenced by clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), and clBeamAudio::ProcessLoop().

7.41.3.6 bool clFreqBeamLine::Get (GDT *, long)

Get data from beamformer.

Parameters:

fpDest Destination vector

lDestCount Number of samples to put into destination vector

Returns:

Success

Definition at line 126 of file FreqBeamLine.cc.

References GDT, lSensorCount, and vSensors.

Referenced by clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), and clBeamAudio::ProcessLoop().

7.41.4 Member Data Documentation

7.41.4.1 long clFreqBeamLine::lSensorCount `[protected]`

Definition at line 41 of file FreqBeamLine.hh.

Referenced by Get(), and Initialize().

7.41.4.2 GDT clFreqBeamLine::fSensorSpacing `[protected]`

Definition at line 42 of file FreqBeamLine.hh.

Referenced by Initialize(), and SetSoundSpeed().

7.41.4.3 std::vector<clArraySensor ∗> clFreqBeamLine::vSensors `[protected]`

Definition at line 43 of file FreqBeamLine.hh.

Referenced by Get(), Initialize(), Put(), SetDebug(), SetDirection(), SetSoundSpeed(), and ∼clFreqBeamLine().

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

- **FreqBeamLine.hh**
- **FreqBeamLine.cc**

7.42 clGraphWidget Class Reference

`#include <GraphWidget.hh>`

Public Member Functions

- **clGraphWidget** ()
- ~**clGraphWidget** ()
- GtkWidget * **Create** (GtkWidget *)
- GtkWidget * **GetPtr** ()
- void **SetSize** (int, int)

Private Member Functions

- void **MapToDisplay** (int *, int *, double, double)

Private Attributes

- long **lWidth**
- long **lHeight**
- long **lMarginTop**
- long **lMarginLeft**
- long **lMarginBottom**
- long **lMarginRight**
- long **lGraphWidth**
- long **lGraphHeight**
- double **dXLeft**
- double **dXRight**
- double **dYBottom**
- double **dYTop**
- double **dXRange**
- double **dYRange**
- GtkWidget * **gwParent**
- GtkWidget * **gwDrawingArea**

7.42.1 Constructor & Destructor Documentation

7.42.1.1 clGraphWidget::clGraphWidget ()

Definition at line 44 of file GraphWidget.cc.

References GRAPHWIDGET_MARGIN_BOTTOM, GRAPHWIDGET_MARGIN_LEFT, GRAPHWIDGET_MARGIN_RIGHT, GRAPHWIDGET_MARGIN_TOP, gwDrawingArea, gwParent, lMarginBottom, lMarginLeft, lMarginRight, and lMarginTop.

7.42.1.2 clGraphWidget::~clGraphWidget ()

Definition at line 56 of file GraphWidget.cc.

7.42.2 Member Function Documentation

7.42.2.1 void clGraphWidget::MapToDisplay (int *, int *, double, double) [private]

Definition at line 29 of file GraphWidget.cc.

References dXLeft, dXRange, dYBottom, dYRange, lGraphHeight, lGraphWidth, lHeight, lMarginBottom, lMarginLeft, and lMarginTop.

7.42.2.2 GtkWidget * clGraphWidget::Create (GtkWidget *)

Definition at line 61 of file GraphWidget.cc.

References gwDrawingArea, and gwParent.

7.42.2.3 GtkWidget* clGraphWidget::GetPtr () [inline]

Definition at line 72 of file GraphWidget.hh.

References gwDrawingArea.

7.42.2.4 void clGraphWidget::SetSize (int, int)

Definition at line 71 of file GraphWidget.cc.

References gwDrawingArea, lGraphHeight, lGraphWidth, lHeight, lMarginBottom, lMarginLeft, lMarginRight, lMarginTop, and lWidth.

7.42.3 Member Data Documentation

7.42.3.1 long clGraphWidget::lWidth [private]

Definition at line 49 of file GraphWidget.hh.

Referenced by SetSize().

7.42.3.2 long clGraphWidget::lHeight [private]

Definition at line 50 of file GraphWidget.hh.

Referenced by MapToDisplay(), and SetSize().

7.42.3.3 long clGraphWidget::lMarginTop [private]

Definition at line 51 of file GraphWidget.hh.

Referenced by clGraphWidget(), MapToDisplay(), and SetSize().

7.42.3.4 long clGraphWidget::lMarginLeft [private]

Definition at line 52 of file GraphWidget.hh.

Referenced by clGraphWidget(), MapToDisplay(), and SetSize().

7.42.3.5 long clGraphWidget::lMarginBottom [private]

Definition at line 53 of file GraphWidget.hh.

Referenced by clGraphWidget(), MapToDisplay(), and SetSize().

7.42.3.6 long clGraphWidget::lMarginRight [private]

Definition at line 54 of file GraphWidget.hh.

Referenced by clGraphWidget(), and SetSize().

7.42.3.7 long clGraphWidget::lGraphWidth [private]

Definition at line 55 of file GraphWidget.hh.

Referenced by MapToDisplay(), and SetSize().

7.42.3.8 long clGraphWidget::lGraphHeight [private]

Definition at line 56 of file GraphWidget.hh.

Referenced by MapToDisplay(), and SetSize().

7.42.3.9 double clGraphWidget::dXLeft [private]

Definition at line 57 of file GraphWidget.hh.

Referenced by MapToDisplay().

7.42.3.10 double clGraphWidget::dXRight [private]

Definition at line 58 of file GraphWidget.hh.

7.42.3.11 double clGraphWidget::dYBottom [private]

Definition at line 59 of file GraphWidget.hh.

Referenced by MapToDisplay().

7.42.3.12 double clGraphWidget::dYTop [private]

Definition at line 60 of file GraphWidget.hh.

7.42.3.13 double clGraphWidget::dXRange [private]

Definition at line 61 of file GraphWidget.hh.

Referenced by MapToDisplay().

7.42.3.14 double clGraphWidget::dYRange [private]

Definition at line 62 of file GraphWidget.hh.

Referenced by MapToDisplay().

7.42.3.15 GtkWidget* clGraphWidget::gwParent [private]

Definition at line 64 of file GraphWidget.hh.

Referenced by clGraphWidget(), and Create().

7.42.3.16 GtkWidget* clGraphWidget::gwDrawingArea [private]

Definition at line 65 of file GraphWidget.hh.

Referenced by clGraphWidget(), Create(), GetPtr(), and SetSize().

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

- **GraphWidget.hh**
- **GraphWidget.cc**

7.43 clGraphWidget::clXGraphWidget Class Reference

Public Member Functions

- **clXGraphWidget** (const char *cpErrorMsg, int iErrorCode=0) throw ()

7.43.1 Constructor & Destructor Documentation

7.43.1.1 clGraphWidget::clXGraphWidget::clXGraphWidget (const char * *cpErrorMsg*, int *iErrorCode* = 0) throw () `[inline]`

Definition at line 43 of file GraphWidget.hh.

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

- **GraphWidget.hh**

7.44 clGtkUtils Class Reference

Gtk+ utility class.

`#include <GtkUtils.hh>`

Public Member Functions

- **clGtkUtils** ()
- ~**clGtkUtils** ()
- void **BuildOptionMenu** (const GtkWidget *, GtkWidget **, GtkWidget **, const char **, int)

  Builds and attaches entries for specified OptionMenu, sets default entry to 0.
- int **OptionMenuGetActive** (const GtkWidget *, GtkWidget **, int)

  Get active item for option menu.
- void **ConnectMotionEvent** (const GtkWidget *, const GtkWidget *)

  Connects widgets motion_notify_event to specified widget.
- void **EnableBackingStore** (const GtkWidget *)

  Enables X backing store for specified widget.
- void **ComboListFromFile** (const GtkWidget *, GList **, const char *)

  Creates Combo widget's popdown menu from specified file.

Private Member Functions

- void **RemoveNewLine** (char *)

7.44.1 Detailed Description

Gtk+ utility class.

Definition at line 41 of file GtkUtils.hh.

7.44.2 Constructor & Destructor Documentation

7.44.2.1 clGtkUtils::clGtkUtils () `[inline]`

Definition at line 45 of file GtkUtils.hh.

7.44.2.2 clGtkUtils::~clGtkUtils () [inline]

Definition at line 46 of file GtkUtils.hh.

7.44.3 Member Function Documentation

7.44.3.1 void clGtkUtils::RemoveNewLine (char *) [inline, private]

Definition at line 39 of file GtkUtils.cc.

Referenced by ComboListFromFile().

7.44.3.2 void clGtkUtils::BuildOptionMenu (const GtkWidget *, GtkWidget **, GtkWidget **, const char **, int)

Builds and attaches entries for specified OptionMenu, sets default entry to 0.

Parameters:

gwOptionMenu Option menu widget (in)

gwpMenu Menu (out)

gwaMenuItem Menu items (out)

cpaMenuTxts Menu item texts (in)

iNumItems Itemcount (in)

Referenced by clSpectGUI::BuildTable2(), clGUILofar::BuildTable2(), clGUILocate::BuildTable2(), clGUILevel::BuildTable2(), clGUIDir::BuildTable2(), clSpectGUI::BuildTable3(), clGUILofar::BuildTable3(), and clGUIDir::BuildTable3().

7.44.3.3 int clGtkUtils::OptionMenuGetActive (const GtkWidget *, GtkWidget **, int)

Get active item for option menu.

Parameters:

gwOptionMenu Option menu

gwaMenuItems Menu items

iMenuItemCount Number of menuitems

Returns:

Index of active menuitem, -1 on error

Referenced by clSpectGUI::GetSettings(), clGUIDir::GetSettings(), clGUILofar::OnPaletteActivate(), clGUILofar::SendSettings(), and clGUILevel::SendSettings().

7.44.3.4 void clGtkUtils::ConnectMotionEvent (const GtkWidget *, const GtkWidget *)

Connects widgets motion_notify_event to specified widget.

This is very useful for connecting drawing area's motion_notify_event to rulers.

Parameters:

gwDest Destination widget

gwSrc Source widget

Definition at line 96 of file GtkUtils.cc.

References EVENT_METHOD.

Referenced by clSpectGUI::ConnectSignals(), clGUILofar::ConnectSignals(), clGUILevel::ConnectSignals(), and clGUIDir::ConnectSignals().

7.44.3.5 void clGtkUtils::EnableBackingStore (const GtkWidget *)

Enables X backing store for specified widget.

Parameters:

gwWidget Widget

Definition at line 106 of file GtkUtils.cc.

Referenced by clGUIDir::Build().

7.44.3.6 void clGtkUtils::ComboListFromFile (const GtkWidget *, GList **, const char *)

Creates Combo widget's popdown menu from specified file.

Note:

Free's list if already allocated

Parameters:

gwWidget Widget

pListPtr GList object pointer (out)

cpFileName Name of file

Definition at line 120 of file GtkUtils.cc.

References GListFreeItem(), GU_TXTBUFSIZE, and RemoveNewLine().

Referenced by clGUITransient::BuildTable1(), clSpectGUI::BuildTable1(), clGUILofar::BuildTable1(), clGUILocate::BuildTable1(), clGUILevel::BuildTable1(), clGUIDir::BuildTable1(), clBeamAudioUI::BuildTable1(), and clSoundChGUI::clSoundChGUI().

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

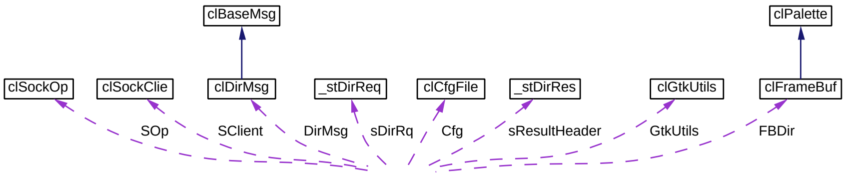
- **GtkUtils.hh**
- **GtkUtils.cc**

7.45 clGUIDir Class Reference

Direction finding GUI.

`#include <GUIDir.hh>`

Collaboration diagram for clGUIDir:



Public Member Functions

- **clGUIDir** (int ∗, char ∗∗∗)
- **~clGUIDir** ()
- int **Exec** ()
- gint **OnDelete** (GtkWidget ∗, GdkEventAny ∗)
- void **OnHideToggled** (GtkToggleButton ∗, gpointer)
- gint **OnConnectClick** (GtkWidget ∗, gpointer gpData)
- void **OnFreezeToggled** (GtkToggleButton ∗, gpointer)
- gint **OnExposeWorm** (GtkWidget ∗, GdkEventExpose ∗)
- gint **OnMotionWorm** (GtkWidget ∗, GdkEventMotion ∗)
- gint **OnPaletteActivate** (GtkWidget ∗, gpointer)
- void **OnSaveClicks** (GtkWidget ∗, gpointer)
- void **OnGdkInput** (gpointer, gint, GdkInputCondition)

Private Member Functions

- bool **Build** ()
- bool **ConnectSignals** ()
- bool **BuildTable1** ()
- bool **BuildTable2** ()
- bool **BuildTable3** ()
- bool **BuildTable4** ()
- bool **BuildTableWorm** ()
- bool **BuildDrawingPrims** ()
- void **FreeDrawingPrims** ()
- bool **ParseServerStr** (char ∗, int ∗, const char ∗)
- bool **InitConnection** (const char ∗, int)
- void **GetSettings** ()
- bool **SendSettings** ()
- void **PrintStatus** ()
- void **SaveInfo** (const char ∗, time_t)
- void **StartNewImgFile** ()

Private Attributes

- volatile bool **bRun**
- volatile bool **bConnected**
- volatile bool **bFreezed**
- bool **bSaving**
- int **iSockH**
- int **iDirectionScale**
- int **iHistoryLines**
- int **iPalette**
- int **iClips**
- int **iWormWidth**
- int **iWormHeight**
- int **iCursorX**
- int **iCursorY**
- int **iTIFFCompression**
- int **iJPEGQuality**
- int **iCompressMode**
- int **iContSaveScans**
- int **iScanCount**
- int **iImgCount**
- long **lResultMsgBufSize**
- long **lResultCount**
- float **fSoundSpeed**
- float **fLeftDirDeg**
- float **fRightDirDeg**
- GDT **fIntegrationTime**
- **stDirReq sDirRq**
- **stDirRes sResultHeader**
- GList * **glServers**
- gint **giGdkTag**
- guint **guSbCtxt**
- GtkWidget * **gwWindow**
- GtkWidget * **gwVBox**
- GtkWidget * **gwCBHide**
- GtkWidget * **gwStatusBar**
- GtkWidget * **gwTable1**
- GtkWidget * **gwLServer**
- GtkWidget * **gwCServer**
- GtkWidget * **gwBConnect**
- GtkWidget * **gwBDisconnect**
- GtkWidget * **gwCBFreeze**
- GtkWidget * **gwTable2**
- GtkWidget * **gwLAlgorithm**
- GtkWidget * **gwOMAlgorithm**
- GtkWidget * **gwMAlgorithm**
- GtkWidget * **gwaMIAlgorithm** [DGUI_ALGORITHM_ITEMS]

- GtkWidget * **gwLSoundSpeed**
- GtkWidget * **gwESoundSpeed**
- GtkWidget * **gwLLowFrequencyLimit**
- GtkWidget * **gwELowFrequencyLimit**
- GtkWidget * **gwLIntegrationTime**
- GtkWidget * **gwEIntegrationTime**
- GtkWidget * **gwTable3**
- GtkWidget * **gwLScaling**
- GtkWidget * **gwOMScaling**
- GtkWidget * **gwMScaling**
- GtkWidget * **gwaMIScaling** [DGUI_SCALING_ITEMS]
- GtkWidget * **gwLScalingExponent**
- GtkWidget * **gwEScalingExponent**
- GtkWidget * **gwLRemoveNoise**
- GtkWidget * **gwOMRemoveNoise**
- GtkWidget * **gwMRemoveNoise**
- GtkWidget * **gwaMIRemoveNoise** [DGUI_REMOVE_NOISE_ITEMS]
- GtkWidget * **gwLAlpha**
- GtkWidget * **gwEAlpha**
- GtkWidget * **gwLMeanLength**
- GtkWidget * **gwEMeanLength**
- GtkWidget * **gwLGapLength**
- GtkWidget * **gwEGapLength**
- GtkWidget * **gwCBNoFilter**
- GtkWidget * **gwCBNormalize**
- GtkWidget * **gwLPalette**
- GtkWidget * **gwOMPalette**
- GtkWidget * **gwMPalette**
- GtkWidget * **gwaMIPalette** [DGUI_PALETTE_ITEMS]
- GtkWidget * **gwTable4**
- GtkWidget * **gwLLeftDirection**
- GtkWidget * **gwELeftDirection**
- GtkWidget * **gwLRightDirection**
- GtkWidget * **gwERightDirection**
- GtkWidget * **gwLSectorCount**
- GtkWidget * **gwESectorCount**
- GtkWidget * **gwLDirectionScale**
- GtkWidget * **gwEDirectionScale**
- GtkWidget * **gwCBSaving**
- GtkWidget * **gwBSave**
- GtkWidget * **gwFSSave**
- GtkWidget * **gwTableWorm**
- GtkWidget * **gwHRDirection**
- GtkWidget * **gwVRTime**
- GtkWidget * **gwDAWorm**
- GdkGC * **ggcWormBG**

- GdkGC * **ggcWormFG**
- GdkCursor * **gcCrossHair**
- std::string **strImgFileName**
- clAlloc **Results**
- clAlloc **ResultMsgBuf**
- clAlloc **ScaledResults**
- **clCfgFile** * **Cfg**
- **clFrameBuf FBDir**
- **clGtkUtils GtkUtils**
- **clSockClie SClient**
- **clSockOp** * **SOp**
- **clDirMsg DirMsg**
- clDSPOp **DSP**

7.45.1 Detailed Description

Direction finding GUI.

Definition at line 77 of file GUIDir.hh.

7.45.2 Constructor & Destructor Documentation

7.45.2.1 clGUIDir::clGUIDir (int *, char ***)

Definition at line 1016 of file GUIDir.cc.

References bConnected, bFreezed, bRun, Cfg, cpWindowTxt, DGUI_CFGFILE, DGUI_PAL_DIR, DGUI_VER_MAJ, DGUI_VER_MIN, DGUI_VER_PL, clCfgFile::GetInt(), glServers, and iPalette.

7.45.2.2 clGUIDir::~clGUIDir ()

Definition at line 1038 of file GUIDir.cc.

References Cfg.

7.45.3 Member Function Documentation

7.45.3.1 bool clGUIDir::Build () `[private]`

Definition at line 147 of file GUIDir.cc.

References BuildDrawingPrims(), BuildTable1(), BuildTable2(), BuildTable3(), BuildTable4(), BuildTableWorm(), cpWindowTxt, DGUI_PADDING, clGtkUtils::EnableBackingStore(), gcCrossHair, GtkUtils, guSbCtxt, gwCBHide, gwDAWorm, gwHRDirection, gwStatusBar, gwVBox, gwVRTime, gwWindow, iDirectionScale, and iHistoryLines.

Referenced by Exec().

7.45.3.2 bool clGUIDir::ConnectSignals () [private]

Definition at line 214 of file GUIDir.cc.

References clGtkUtils::ConnectMotionEvent(), DGUI_PALETTE_ITEMS, GtkUtils, gwaMIPalette, gwBConnect, gwBDisconnect, gwBSave, gwCBFreeze, gwCBHide, gwCBSaving, gwDAWorm, gwFSSave, gwHRDirection, gwVRTime, gwWindow, WrapOnConnectClick(), WrapOnDelete(), WrapOnExposeWorm(), WrapOnFreezeToggled(), WrapOnHideToggled(), WrapOnMotionWorm(), WrapOnPaletteActivate(), and WrapOnSaveClicks().

Referenced by Exec().

7.45.3.3 bool clGUIDir::BuildTable1 () [private]

Definition at line 258 of file GUIDir.cc.

References clGtkUtils::ComboListFromFile(), cpBConnectTxt, cpBDisconnectTxt, cpCBFreezeTxt, cpLServerTxt, DGUI_HOSTFILE, DGUI_PADDING, DGUI_SERVER_MAXLEN, glServers, GtkUtils, gwBConnect, gwBDisconnect, gwCBFreeze, gwCServer, gwLServer, gwTable1, and gwVBox.

Referenced by Build().

7.45.3.4 bool clGUIDir::BuildTable2 () [private]

Definition at line 318 of file GUIDir.cc.

References clGtkUtils::BuildOptionMenu(), cpaLAlgorithmMenu, cpLAlgorithmTxt, cpLIntegrationTimeTxt, cpLLowFrequencyLimitTxt, cpLSoundSpeedTxt, DGUI_ALGORITHM_ITEMS, DGUI_CONV_BUF_SIZE, DGUI_ENTRY_WIDTH, DGUI_PADDING, fSoundSpeed, GtkUtils, gwaMIAlgorithm, gwEIntegrationTime, gwELowFrequencyLimit, gwESoundSpeed, gwLAlgorithm, gwLIntegrationTime, gwLLowFrequencyLimit, gwLSoundSpeed, gwMAlgorithm, gwOMAlgorithm, gwTable2, and gwVBox.

Referenced by Build().

7.45.3.5 bool clGUIDir::BuildTable3 () [private]

Definition at line 410 of file GUIDir.cc.

References clGtkUtils::BuildOptionMenu(), cpaLPaletteMenu, cpaLRemoveNoiseMenu, cpaLScalingMenu, cpCBNoFilterTxt, cpCBNormalizeTxt, cpLAlphaTxt, cpLGapLengthTxt, cpLMeanLengthTxt, cpLPaletteTxt, cpLRemoveNoiseTxt, cpLScalingExponentTxt, cpLScalingTxt, DGUI_ENTRY_WIDTH, DGUI_PADDING, DGUI_PALETTE_ITEMS, DGUI_REMOVE_NOISE_ITEMS, DGUI_SCALING_ITEMS, GtkUtils, gwaMIPalette, gwaMIRemoveNoise, gwaMIScaling, gwCBNoFilter, gwCBNormalize, gwEAlpha, gwEGapLength, gwEMeanLength, gwEScalingExponent, gwLAlpha, gwLGapLength, gwLMeanLength, gwLPalette, gwLRemoveNoise, gwLScaling, gwLScalingExponent, gwMPalette, gwMRemoveNoise, gwMScaling, gwOMPalette, gwOMRemoveNoise, gwOMScaling, gwTable3, gwVBox, and iPalette.

Referenced by Build().

7.45.3.6 bool clGUIDir::BuildTable4 () [private]

Definition at line 571 of file GUIDir.cc.

References cpBSaveTxt, cpCBSavingTxt, cpFSSaveTxt, cpLDirectionScaleTxt, cpLLeftDirectionTxt, cpLRightDirectionTxt, cpLSectorCountTxt, DGUI_CONV_BUF_SIZE, DGUI_ENTRY_WIDTH, DGUI_PADDING, gwBSave, gwCBSaving, gwEDirectionScale, gwELeftDirection, gwERightDirection, gwESectorCount, gwFSSave, gwLDirectionScale, gwLLeftDirection, gwLRightDirection, gwLSectorCount, gwTable4, gwVBox, and iDirectionScale.

Referenced by Build().

7.45.3.7 bool clGUIDir::BuildTableWorm () [private]

Definition at line 684 of file GUIDir.cc.

References gwDAWorm, gwHRDirection, gwTableWorm, gwVBox, gwVRTime, iDirectionScale, and iHistoryLines.

Referenced by Build().

7.45.3.8 bool clGUIDir::BuildDrawingPrims () [private]

Definition at line 733 of file GUIDir.cc.

References DGUI_PAL_BW, DGUI_PAL_DIR, DGUI_PAL_GREEN, DGUI_PAL_GREEN2, DGUI_PAL_HSV, DGUI_PAL_LIGHT, DGUI_PAL_PUREGREEN, DGUI_PAL_TEMP, DGUI_PAL_WB, DGUI_WORM_BG, DGUI_WORM_FG, FBDir, gcCrossHair, ggcWormBG, ggcWormFG, gwDAWorm, iPalette, clFrameBuf::PalGenBW(), clFrameBuf::PalGenDir(), clFrameBuf::PalGenGreen(), clFrameBuf::PalGenGreen2(), clFrameBuf::PalGenHSV(), clFrameBuf::PalGenLight(), clFrameBuf::PalGenPureGreen(), clFrameBuf::PalGenTemp(), and clFrameBuf::PalGenWB().

Referenced by Build().

7.45.3.9 void clGUIDir::FreeDrawingPrims () [private]

Definition at line 789 of file GUIDir.cc.

References gcCrossHair, ggcWormBG, and ggcWormFG.

Referenced by Exec().

7.45.3.10 bool clGUIDir::ParseServerStr (char *, int *, const char *) [private]

Definition at line 797 of file GUIDir.cc.

References DGUI_SERVER_MAXLEN.

Referenced by OnConnectClick().

7.45.3.11 bool clGUIDir::InitConnection (const char *, int) [private]

Definition at line 820 of file GUIDir.cc.

References bConnected, clSockClie::Connect(), DGUI_REQ_PROC, DGUI_REQ_PROC2, FBDir, GDT, clFrameBuf::GetCurPtr(), clFrameBuf::GetRowStride(), GetSettings(), ggcWormFG, giGdkTag, GLOBAL_HEADER_LEN, gwDAWorm, gwHRDirection, gwVRTime, _stDirReq::iAlgorithm, iClips, iDirectionScale, iHistoryLines, iSockH, iWormHeight, iWormWidth, lResultCount, lResultMsgBufSize, MSG_DIR_ALG_BEAM, MSG_DIR_ALG_CORR, MSG_DIR_ALG_SPECT, ResultMsgBuf, Results, ScaledResults, SClient, sDirRq, SendSettings(), clFrameBuf::SetSize(), SOp, WrapOnGdkInput(), and clSockOp::WriteN().

Referenced by OnConnectClick().

7.45.3.12 void clGUIDir::GetSettings () [private]

Definition at line 868 of file GUIDir.cc.

References _stDirReq::bDisableFilter, _stDirReq::bNormalize, DGUI_ALGORITHM_ITEMS, DGUI_REMOVE_NOISE_ITEMS, DGUI_SCALING_ITEMS, DSP, _stDirReq::fAlpha, _stDirReq::fIntegrationTime, fIntegrationTime, _stDirReq::fLeftDir, fLeftDirDeg, _stDirReq::fLowFreqLimit, _stDirReq::fRightDir, fRightDirDeg, _stDirReq::fScalingExp, _stDirReq::fSoundSpeed, GtkUtils, gwaMIAlgorithm, gwaMIRemoveNoise, gwaMIScaling, gwCBNoFilter, gwCBNormalize, gwEAlpha, gwEDirectionScale, gwEGapLength, gwEIntegrationTime, gwELeftDirection, gwELowFrequencyLimit, gwEMeanLength, gwERightDirection, gwEScalingExponent, gwESectorCount, gwESoundSpeed, gwHRDirection, gwOMAlgorithm, gwOMRemoveNoise, gwOMScaling, gwVRTime, _stDirReq::iAlgorithm, iDirectionScale, iHistoryLines, _stDirReq::iRemoveNoise, _stDirReq::iScaling, _stDirReq::lGapLength, _stDirReq::lMeanLength, lResultCount, _stDirReq::lSectorCount, MSG_DIR_ALG_BEAM, MSG_DIR_ALG_CORR, MSG_DIR_ALG_SPECT, clGtkUtils::OptionMenuGetActive(), and sDirRq.

Referenced by InitConnection().

7.45.3.13 bool clGUIDir::SendSettings () [private]

Definition at line 930 of file GUIDir.cc.

References DirMsg, clSockOp::GetErrno(), GLOBAL_HEADER_LEN, sDirRq, clDirMsg::SetRequest(), SOp, and clSockOp::WriteN().

Referenced by InitConnection().

7.45.3.14 void clGUIDir::PrintStatus () [private]

Definition at line 945 of file GUIDir.cc.

References DGUI_CONV_BUF_SIZE, fIntegrationTime, fLeftDirDeg, _stDirRes::fPeakLevel, fRightDirDeg, guSbCtxt, gwStatusBar, iClips, iCursorX, iCursorY, iDirectionScale, _stDirRes::lSectorCount, and sResultHeader.

Referenced by OnGdkInput(), and OnMotionWorm().

7.45.3.15 void clGUIDir::SaveInfo (const char *, time_t) [private]

Definition at line 976 of file GUIDir.cc.

References FB_TIFF_MODE, _stDirRes::fHighFreqLimit, _stDirRes::fIntegrationTime, _stDirReq::fLeftDir, _stDirReq::fRightDir, _stDirRes::lSectorCount, sDirRq, and sResultHeader.

Referenced by OnSaveClicks(), and StartNewImgFile().

7.45.3.16 void clGUIDir::StartNewImgFile () [private]

Definition at line 997 of file GUIDir.cc.

References bSaving, FB_TIFF_CONT_VERTICAL, FBDir, gwCBSaving, iCompressMode, iImgCount, iJPEGQuality, SaveInfo(), clFrameBuf::StartSaveToFile(), clFrameBuf::StopSaveToFile(), and strImgFileName.

Referenced by OnGdkInput().

7.45.3.17 int clGUIDir::Exec ()

Definition at line 1044 of file GUIDir.cc.

References Build(), Cfg, clFrameBuf::Clear(), ConnectSignals(), DGUI_DEF_LINES, DGUI_DEF_SOUNDSPEED, FB_TIFF_COMPRESS_DEFLATE, FB_TIFF_COMPRESS_JPEG, FB_TIFF_COMPRESS_LZW, FB_TIFF_COMPRESS_NONE, FB_TIFF_COMPRESS_RLE, FBDir, FreeDrawingPrims(), fSoundSpeed, clCfgFile::GetFlt(), clCfgFile::GetInt(), iCompressMode, iContSaveScans, iDirectionScale, iHistoryLines, iJPEGQuality, iPalette, iTIFFCompression, iWormHeight, iWormWidth, and clFrameBuf::SetSize().

Referenced by main().

7.45.3.18 gint clGUIDir::OnDelete (GtkWidget ∗, GdkEventAny ∗)

Definition at line 1100 of file GUIDir.cc.

References bRun, bSaving, FBDir, and clFrameBuf::StopSaveToFile().

Referenced by WrapOnDelete().

7.45.3.19 void clGUIDir::OnHideToggled (GtkToggleButton ∗, gpointer)

Definition at line 1112 of file GUIDir.cc.

References gwTable1, gwTable2, gwTable3, and gwTable4.

Referenced by WrapOnHideToggled().

7.45.3.20 gint clGUIDir::OnConnectClick (GtkWidget ∗, gpointer *gpData*)

Definition at line 1131 of file GUIDir.cc.

References bConnected, DGUI_SERVER_MAXLEN, giGdkTag, gwBConnect, gwCServer, InitConnection(), ParseServerStr(), clSockOp::Shutdown(), and SOp.

Referenced by WrapOnConnectClick().

7.45.3.21 void clGUIDir::OnFreezeToggled (GtkToggleButton ∗, gpointer)

Definition at line 1169 of file GUIDir.cc.

References bFreezed.

Referenced by WrapOnFreezeToggled().

7.45.3.22 gint clGUIDir::OnExposeWorm (GtkWidget ∗, GdkEventExpose ∗)

Definition at line 1175 of file GUIDir.cc.

References FBDir, clFrameBuf::GetCurPtr(), clFrameBuf::GetRowStride(), ggcWormFG, gwDAWorm, iWormHeight, and iWormWidth.

Referenced by WrapOnExposeWorm().

7.45.3.23 gint clGUIDir::OnMotionWorm (GtkWidget ∗, GdkEventMotion ∗)

Definition at line 1196 of file GUIDir.cc.

References iCursorX, iCursorY, and PrintStatus().

Referenced by WrapOnMotionWorm().

7.45.3.24 gint clGUIDir::OnPaletteActivate (GtkWidget *, gpointer)

Definition at line 1205 of file GUIDir.cc.

References Cfg, DGUI_PAL_BW, DGUI_PAL_DIR, DGUI_PAL_GREEN, DGUI_PAL_GREEN2, DGUI_PAL_HSV, DGUI_PAL_LIGHT, DGUI_PAL_PUREGREEN, DGUI_PAL_TEMP, DGUI_PAL_WB, FBDir, clCfgFile::GetInt(), gwaMIPalette, clFrameBuf::PalGenBW(), clFrameBuf::PalGenDir(), clFrameBuf::PalGenGreen(), clFrameBuf::PalGenGreen2(), clFrameBuf::PalGenHSV(), clFrameBuf::PalGenLight(), clFrameBuf::PalGenPureGreen(), clFrameBuf::PalGenTemp(), and clFrameBuf::PalGenWB().

Referenced by WrapOnPaletteActivate().

7.45.3.25 void clGUIDir::OnSaveClicks (GtkWidget *, gpointer)

Definition at line 1277 of file GUIDir.cc.

References bSaving, FB_TIFF_CONT_VERTICAL, FBDir, _stDirRes::fIntegrationTime, clFrameBuf::GetHeight(), gwCBSaving, gwFSSave, iCompressMode, iImgCount, iJPEGQuality, iScanCount, SaveInfo(), clFrameBuf::SaveToFile(), sResultHeader, clFrameBuf::StartSaveToFile(), clFrameBuf::StopSaveToFile(), and strImgFileName.

Referenced by WrapOnSaveClicks().

7.45.3.26 void clGUIDir::OnGdkInput (gpointer, gint, GdkInputCondition)

Definition at line 1361 of file GUIDir.cc.

References bConnected, bFreezed, bRun, bSaving, DGUI_CONV_BUF_SIZE, DirMsg, clFrameBuf::DrawLine(), DSP, FBDir, _stDirRes::fIntegrationTime, fIntegrationTime, _stDirRes::fPeakLevel, GDT, clFrameBuf::GetCurPtr(), clSockOp::GetErrno(), clDirMsg::GetResult(), clFrameBuf::GetRowStride(), ggcWormFG, giGdkTag, gwDAWorm, gwEIntegrationTime, gwVRTime, iClips, iContSaveScans, iDirectionScale, iHistoryLines, iImgCount, iScanCount, iWormHeight, iWormWidth, lResultCount, lResultMsgBufSize, _stDirRes::lSectorCount, PrintStatus(), clSockOp::ReadN(), ResultMsgBuf, Results, ScaledResults, clSockOp::Shutdown(), SOp, sResultHeader, and StartNewImgFile().

Referenced by WrapOnGdkInput().

7.45.4 Member Data Documentation

7.45.4.1 volatile bool clGUIDir::bRun [private]

Definition at line 80 of file GUIDir.hh.

Referenced by clGUIDir(), OnDelete(), and OnGdkInput().

7.45.4.2 volatile bool clGUIDir::bConnected [private]

Definition at line 81 of file GUIDir.hh.

Referenced by clGUIDir(), InitConnection(), OnConnectClick(), and OnGdkInput().

7.45.4.3 volatile bool clGUIDir::bFreezed [private]

Definition at line 82 of file GUIDir.hh.

Referenced by clGUIDir(), OnFreezeToggled(), and OnGdkInput().

7.45.4.4 bool clGUIDir::bSaving [private]

Definition at line 83 of file GUIDir.hh.

Referenced by OnDelete(), OnGdkInput(), OnSaveClicks(), and StartNewImgFile().

7.45.4.5 int clGUIDir::iSockH [private]

Definition at line 84 of file GUIDir.hh.

Referenced by InitConnection().

7.45.4.6 int clGUIDir::iDirectionScale [private]

Definition at line 85 of file GUIDir.hh.

Referenced by Build(), BuildTable4(), BuildTableWorm(), Exec(), GetSettings(), InitConnection(), OnGdkInput(), and PrintStatus().

7.45.4.7 int clGUIDir::iHistoryLines [private]

Definition at line 86 of file GUIDir.hh.

Referenced by Build(), BuildTableWorm(), Exec(), GetSettings(), InitConnection(), and OnGdkInput().

7.45.4.8 int clGUIDir::iPalette [private]

Definition at line 87 of file GUIDir.hh.

Referenced by BuildDrawingPrims(), BuildTable3(), clGUIDir(), and Exec().

7.45.4.9 int clGUIDir::iClips [private]

Definition at line 88 of file GUIDir.hh.

Referenced by InitConnection(), OnGdkInput(), and PrintStatus().

7.45.4.10 int clGUIDir::iWormWidth [private]

Definition at line 89 of file GUIDir.hh.

Referenced by Exec(), InitConnection(), OnExposeWorm(), and OnGdkInput().

7.45.4.11 int clGUIDir::iWormHeight [private]

Definition at line 90 of file GUIDir.hh.

Referenced by Exec(), InitConnection(), OnExposeWorm(), and OnGdkInput().

7.45.4.12 int clGUIDir::iCursorX [private]

Definition at line 91 of file GUIDir.hh.

Referenced by OnMotionWorm(), and PrintStatus().

7.45.4.13 int clGUIDir::iCursorY [private]

Definition at line 92 of file GUIDir.hh.

Referenced by OnMotionWorm(), and PrintStatus().

7.45.4.14 int clGUIDir::iTIFFCompression [private]

Definition at line 93 of file GUIDir.hh.

Referenced by Exec().

7.45.4.15 int clGUIDir::iJPEGQuality [private]

Definition at line 94 of file GUIDir.hh.

Referenced by Exec(), OnSaveClicks(), and StartNewImgFile().

7.45.4.16 int clGUIDir::iCompressMode [private]

Definition at line 95 of file GUIDir.hh.

Referenced by Exec(), OnSaveClicks(), and StartNewImgFile().

7.45.4.17 int clGUIDir::iContSaveScans [private]

Definition at line 96 of file GUIDir.hh.

Referenced by Exec(), and OnGdkInput().

7.45.4.18 int clGUIDir::iScanCount [private]

Definition at line 97 of file GUIDir.hh.

Referenced by OnGdkInput(), and OnSaveClicks().

7.45.4.19 int clGUIDir::iImgCount [private]

Definition at line 98 of file GUIDir.hh.

Referenced by OnGdkInput(), OnSaveClicks(), and StartNewImgFile().

7.45.4.20 long clGUIDir::lResultMsgBufSize [private]

Definition at line 99 of file GUIDir.hh.

Referenced by InitConnection(), and OnGdkInput().

7.45.4.21 long clGUIDir::lResultCount [private]

Definition at line 100 of file GUIDir.hh.

Referenced by GetSettings(), InitConnection(), and OnGdkInput().

7.45.4.22 float clGUIDir::fSoundSpeed [private]

Definition at line 101 of file GUIDir.hh.

Referenced by BuildTable2(), and Exec().

7.45.4.23 float clGUIDir::fLeftDirDeg [private]

Definition at line 102 of file GUIDir.hh.

Referenced by GetSettings(), and PrintStatus().

7.45.4.24 float clGUIDir::fRightDirDeg [private]

Definition at line 103 of file GUIDir.hh.

Referenced by GetSettings(), and PrintStatus().

7.45.4.25 GDT clGUIDir::fIntegrationTime [private]

Definition at line 104 of file GUIDir.hh.

Referenced by GetSettings(), OnGdkInput(), and PrintStatus().

7.45.4.26 stDirReq clGUIDir::sDirRq [private]

Definition at line 105 of file GUIDir.hh.

Referenced by GetSettings(), InitConnection(), SaveInfo(), and SendSettings().

7.45.4.27 stDirRes clGUIDir::sResultHeader [private]

Definition at line 106 of file GUIDir.hh.

Referenced by OnGdkInput(), OnSaveClicks(), PrintStatus(), and SaveInfo().

7.45.4.28 GList* clGUIDir::glServers [private]

Definition at line 108 of file GUIDir.hh.

Referenced by BuildTable1(), and clGUIDir().

7.45.4.29 gint clGUIDir::giGdkTag [private]

Definition at line 110 of file GUIDir.hh.

Referenced by InitConnection(), OnConnectClick(), and OnGdkInput().

7.45.4.30 guint clGUIDir::guSbCtxt [private]

Definition at line 111 of file GUIDir.hh.

Referenced by Build(), and PrintStatus().

7.45.4.31 GtkWidget* clGUIDir::gwWindow [private]

Definition at line 113 of file GUIDir.hh.

Referenced by Build(), and ConnectSignals().

7.45.4.32 GtkWidget* clGUIDir::gwVBox [private]

Definition at line 114 of file GUIDir.hh.

Referenced by Build(), BuildTable1(), BuildTable2(), BuildTable3(), BuildTable4(), and BuildTableWorm().

7.45.4.33 GtkWidget* clGUIDir::gwCBHide [private]

Definition at line 115 of file GUIDir.hh.

Referenced by Build(), and ConnectSignals().

7.45.4.34 GtkWidget* clGUIDir::gwStatusBar [private]

Definition at line 116 of file GUIDir.hh.

Referenced by Build(), and PrintStatus().

7.45.4.35 GtkWidget* clGUIDir::gwTable1 [private]

Definition at line 118 of file GUIDir.hh.

Referenced by BuildTable1(), and OnHideToggled().

7.45.4.36 GtkWidget* clGUIDir::gwLServer [private]

Definition at line 119 of file GUIDir.hh.

Referenced by BuildTable1().

7.45.4.37 GtkWidget* clGUIDir::gwCServer [private]

Definition at line 120 of file GUIDir.hh.

Referenced by BuildTable1(), and OnConnectClick().

7.45.4.38 GtkWidget* clGUIDir::gwBConnect [private]

Definition at line 121 of file GUIDir.hh.

Referenced by BuildTable1(), ConnectSignals(), and OnConnectClick().

7.45.4.39 GtkWidget* clGUIDir::gwBDisconnect [private]

Definition at line 122 of file GUIDir.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.45.4.40 GtkWidget* clGUIDir::gwCBFreeze [private]

Definition at line 123 of file GUIDir.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.45.4.41 GtkWidget* clGUIDir::gwTable2 [private]

Definition at line 125 of file GUIDir.hh.

Referenced by BuildTable2(), and OnHideToggled().

7.45.4.42 GtkWidget* clGUIDir::gwLAlgorithm [private]

Definition at line 126 of file GUIDir.hh.

Referenced by BuildTable2().

7.45.4.43 GtkWidget* clGUIDir::gwOMAlgorithm [private]

Definition at line 127 of file GUIDir.hh.

Referenced by BuildTable2(), and GetSettings().

7.45.4.44 GtkWidget* clGUIDir::gwMAlgorithm [private]

Definition at line 128 of file GUIDir.hh.

Referenced by BuildTable2().

7.45.4.45 GtkWidget* clGUIDir::gwaMIAlgorithm[DGUI_ALGORITHM_ITEMS] [private]

Definition at line 129 of file GUIDir.hh.

Referenced by BuildTable2(), and GetSettings().

7.45.4.46 GtkWidget* clGUIDir::gwLSoundSpeed [private]

Definition at line 130 of file GUIDir.hh.

Referenced by BuildTable2().

7.45.4.47 GtkWidget* clGUIDir::gwESoundSpeed [private]

Definition at line 131 of file GUIDir.hh.

Referenced by BuildTable2(), and GetSettings().

7.45.4.48 GtkWidget* clGUIDir::gwLLowFrequencyLimit [private]

Definition at line 132 of file GUIDir.hh.

Referenced by BuildTable2().

7.45.4.49 GtkWidget* clGUIDir::gwELowFrequencyLimit [private]

Definition at line 133 of file GUIDir.hh.

Referenced by BuildTable2(), and GetSettings().

7.45.4.50 GtkWidget* clGUIDir::gwLIntegrationTime [private]

Definition at line 134 of file GUIDir.hh.

Referenced by BuildTable2().

7.45.4.51 GtkWidget* clGUIDir::gwEIntegrationTime [private]

Definition at line 135 of file GUIDir.hh.

Referenced by BuildTable2(), GetSettings(), and OnGdkInput().

7.45.4.52 GtkWidget* clGUIDir::gwTable3 [private]

Definition at line 137 of file GUIDir.hh.

Referenced by BuildTable3(), and OnHideToggled().

7.45.4.53 GtkWidget* clGUIDir::gwLScaling [private]

Definition at line 138 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.54 GtkWidget* clGUIDir::gwOMScaling [private]

Definition at line 139 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.55 GtkWidget* clGUIDir::gwMScaling [private]

Definition at line 140 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.56 GtkWidget* clGUIDir::gwaMIScaling[DGUI_SCALING_ITEMS] [private]

Definition at line 141 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.57 GtkWidget* clGUIDir::gwLScalingExponent [private]

Definition at line 142 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.58 GtkWidget* clGUIDir::gwEScalingExponent [private]

Definition at line 143 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.59 GtkWidget* clGUIDir::gwLRemoveNoise [private]

Definition at line 144 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.60 GtkWidget* clGUIDir::gwOMRemoveNoise [private]

Definition at line 145 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.61 GtkWidget* clGUIDir::gwMRemoveNoise [private]

Definition at line 146 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.62 GtkWidget* clGUIDir::gwaMIRemoveNoise[DGUI_REMOVE_NOISE_ITEMS] [private]

Definition at line 147 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.63 GtkWidget* clGUIDir::gwLAlpha [private]

Definition at line 148 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.64 GtkWidget* clGUIDir::gwEAlpha [private]

Definition at line 149 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.65 GtkWidget* clGUIDir::gwLMeanLength [private]

Definition at line 150 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.66 GtkWidget* clGUIDir::gwEMeanLength [private]

Definition at line 151 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.67 GtkWidget* clGUIDir::gwLGapLength [private]

Definition at line 152 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.68 GtkWidget* clGUIDir::gwEGapLength [private]

Definition at line 153 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.69 GtkWidget* clGUIDir::gwCBNoFilter [private]

Definition at line 154 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.70 GtkWidget* clGUIDir::gwCBNormalize [private]

Definition at line 155 of file GUIDir.hh.

Referenced by BuildTable3(), and GetSettings().

7.45.4.71 GtkWidget* clGUIDir::gwLPalette [private]

Definition at line 156 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.72 GtkWidget* clGUIDir::gwOMPalette [private]

Definition at line 157 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.73 GtkWidget* clGUIDir::gwMPalette [private]

Definition at line 158 of file GUIDir.hh.

Referenced by BuildTable3().

7.45.4.74 GtkWidget* clGUIDir::gwaMIPalette[DGUI_PALETTE_ITEMS] [private]

Definition at line 159 of file GUIDir.hh.

Referenced by BuildTable3(), ConnectSignals(), and OnPaletteActivate().

7.45.4.75 GtkWidget* clGUIDir::gwTable4 [private]

Definition at line 161 of file GUIDir.hh.

Referenced by BuildTable4(), and OnHideToggled().

7.45.4.76 GtkWidget* clGUIDir::gwLLeftDirection [private]

Definition at line 162 of file GUIDir.hh.

Referenced by BuildTable4().

7.45.4.77 GtkWidget* clGUIDir::gwELeftDirection [private]

Definition at line 163 of file GUIDir.hh.

Referenced by BuildTable4(), and GetSettings().

7.45.4.78 GtkWidget* clGUIDir::gwLRightDirection [private]

Definition at line 164 of file GUIDir.hh.

Referenced by BuildTable4().

7.45.4.79 GtkWidget* clGUIDir::gwERightDirection [private]

Definition at line 165 of file GUIDir.hh.

Referenced by BuildTable4(), and GetSettings().

7.45.4.80 GtkWidget* clGUIDir::gwLSectorCount [private]

Definition at line 166 of file GUIDir.hh.

Referenced by BuildTable4().

7.45.4.81 GtkWidget* clGUIDir::gwESectorCount [private]

Definition at line 167 of file GUIDir.hh.

Referenced by BuildTable4(), and GetSettings().

7.45.4.82 GtkWidget* clGUIDir::gwLDirectionScale [private]

Definition at line 168 of file GUIDir.hh.

Referenced by BuildTable4().

7.45.4.83 GtkWidget* clGUIDir::gwEDirectionScale [private]

Definition at line 169 of file GUIDir.hh.

Referenced by BuildTable4(), and GetSettings().

7.45.4.84 GtkWidget* clGUIDir::gwCBSaving [private]

Definition at line 170 of file GUIDir.hh.

Referenced by BuildTable4(), ConnectSignals(), OnSaveClicks(), and StartNew-ImgFile().

7.45.4.85 GtkWidget* clGUIDir::gwBSave [private]

Definition at line 171 of file GUIDir.hh.

Referenced by BuildTable4(), and ConnectSignals().

7.45.4.86 GtkWidget* clGUIDir::gwFSSave [private]

Definition at line 172 of file GUIDir.hh.

Referenced by BuildTable4(), ConnectSignals(), and OnSaveClicks().

7.45.4.87 GtkWidget* clGUIDir::gwTableWorm [private]

Definition at line 174 of file GUIDir.hh.

Referenced by BuildTableWorm().

7.45.4.88 GtkWidget* clGUIDir::gwHRDirection [private]

Definition at line 175 of file GUIDir.hh.

Referenced by Build(), BuildTableWorm(), ConnectSignals(), GetSettings(), and InitConnection().

7.45.4.89 GtkWidget* clGUIDir::gwVRTime [private]

Definition at line 176 of file GUIDir.hh.

Referenced by Build(), BuildTableWorm(), ConnectSignals(), GetSettings(), InitConnection(), and OnGdkInput().

7.45.4.90 GtkWidget* clGUIDir::gwDAWorm [private]

Definition at line 177 of file GUIDir.hh.

Referenced by Build(), BuildDrawingPrims(), BuildTableWorm(), ConnectSignals(), InitConnection(), OnExposeWorm(), and OnGdkInput().

7.45.4.91 GdkGC* clGUIDir::ggcWormBG [private]

Definition at line 179 of file GUIDir.hh.

Referenced by BuildDrawingPrims(), and FreeDrawingPrims().

7.45.4.92 GdkGC* clGUIDir::ggcWormFG [private]

Definition at line 180 of file GUIDir.hh.

Referenced by BuildDrawingPrims(), FreeDrawingPrims(), InitConnection(), OnExposeWorm(), and OnGdkInput().

7.45.4.93 GdkCursor* clGUIDir::gcCrossHair [private]

Definition at line 181 of file GUIDir.hh.

Referenced by Build(), BuildDrawingPrims(), and FreeDrawingPrims().

7.45.4.94 std::string clGUIDir::strImgFileName [private]

Definition at line 183 of file GUIDir.hh.

Referenced by OnSaveClicks(), and StartNewImgFile().

7.45.4.95 clAlloc clGUIDir::Results [private]

Definition at line 184 of file GUIDir.hh.

Referenced by InitConnection(), and OnGdkInput().

7.45.4.96 clAlloc clGUIDir::ResultMsgBuf [private]

Definition at line 185 of file GUIDir.hh.

Referenced by InitConnection(), and OnGdkInput().

7.45.4.97 clAlloc clGUIDir::ScaledResults [private]

Definition at line 186 of file GUIDir.hh.

Referenced by InitConnection(), and OnGdkInput().

7.45.4.98 clCfgFile* clGUIDir::Cfg [private]

Definition at line 187 of file GUIDir.hh.

Referenced by clGUIDir(), Exec(), OnPaletteActivate(), and ~clGUIDir().

7.45.4.99 clFrameBuf clGUIDir::FBDir [private]

Definition at line 188 of file GUIDir.hh.

Referenced by BuildDrawingPrims(), Exec(), InitConnection(), OnDelete(), OnExposeWorm(), OnGdkInput(), OnPaletteActivate(), OnSaveClicks(), and StartNewImgFile().

7.45.4.100 clGtkUtils clGUIDir::GtkUtils [private]

Definition at line 189 of file GUIDir.hh.

Referenced by Build(), BuildTable1(), BuildTable2(), BuildTable3(), ConnectSignals(), and GetSettings().

7.45.4.101 clSockClie clGUIDir::SClient [private]

Definition at line 190 of file GUIDir.hh.

Referenced by InitConnection().

7.45.4.102 clSockOp* clGUIDir::SOp [private]

Definition at line 191 of file GUIDir.hh.

Referenced by InitConnection(), OnConnectClick(), OnGdkInput(), and SendSettings().

7.45.4.103 clDirMsg clGUIDir::DirMsg [private]

Definition at line 192 of file GUIDir.hh.

Referenced by OnGdkInput(), and SendSettings().

7.45.4.104 clDSPOp clGUIDir::DSP [private]

Definition at line 193 of file GUIDir.hh.

Referenced by GetSettings(), and OnGdkInput().

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

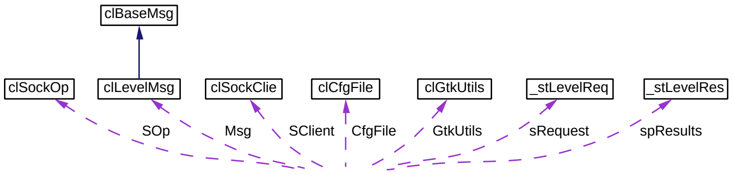
- **GUIDir.hh**
- **GUIDir.cc**

7.46 clGUILevel Class Reference

GUI for level server.

`#include <GUILevel.hh>`

Collaboration diagram for clGUILevel:



Public Member Functions

- **clGUILevel** ()
- ~**clGUILevel** ()
- int **Main** (int ∗, char ∗∗∗)
- gboolean **OnDelete** (GtkWidget ∗, GdkEvent ∗, gpointer)
- void **OnConnectClick** (GtkButton ∗, gpointer)
- gboolean **OnSofarExpose** (GtkWidget ∗, GdkEventExpose ∗, gpointer)
- gboolean **OnSofarMotion** (GtkWidget ∗, GdkEventMotion ∗, gpointer)
- gboolean **OnSofarConfigure** (GtkWidget ∗, GdkEventConfigure ∗, gpointer)
- void **OnGdkInput** (gpointer, gint, GdkInputCondition)

Private Member Functions

- bool **GetCfg** ()
- bool **Build** ()
- bool **BuildTable1** ()
- bool **BuildTable2** ()
- bool **BuildSofar** ()
- bool **ConnectSignals** ()
- bool **BuildDrawingPrims** ()
- bool **ConnectToServer** (const char ∗, int)
- bool **SendSettings** ()
- void **DisplayResults** ()
- void **InitializeDisplay** ()

Private Attributes

- bool **bRun**
- bool **bConnected**
- bool **bFirstResult**
- int **iBeamCount**
- long **lResultPos**
- long **lResultCount**
- float **fDisplayLow**
- float **fDisplayHigh**
- char **cpResultBuf** [GLOBAL_HEADER_LEN]
- **stLevelReq sRequest**
- **stpLevelRes spResults**
- guint **guSbCtxt**
- gint **iGdkInputTag**
- GList * **glServer**
- GdkGC * **ggcSofarBG**
- GdkGC * **ggcSofarFG**
- GdkCursor * **gcCrossHair**
- GtkWidget * **gwWindow**
- GtkWidget * **gwVBox**
- GtkWidget * **gwStatusBar**
- GtkWidget * **gwTable1**
- GtkWidget * **gwLServer**
- GtkWidget * **gwCServer**
- GtkWidget * **gwLChannel**
- GtkObject * **goAChannel**
- GtkWidget * **gwSBChannel**
- GtkWidget * **gwBConnect**
- GtkWidget * **gwTable2**
- GtkWidget * **gwLAlgorithm**
- GtkWidget * **gwOMAlgorithm**
- GtkWidget * **gwMAlgorithm**
- GtkWidget * **gwaMIAlgorithm** [GUILEV_ALGORITHM_ITEMS]
- GtkWidget * **gwLIntegrationTime**
- GtkWidget * **gwEIntegrationTime**
- GtkWidget * **gwLLowFrequency**
- GtkWidget * **gwELowFrequency**
- GtkWidget * **gwLHighFrequency**
- GtkWidget * **gwEHighFrequency**
- GtkWidget * **gwLDisplayLow**
- GtkWidget * **gwEDisplayLow**
- GtkWidget * **gwLDisplayHigh**
- GtkWidget * **gwEDisplayHigh**
- GtkWidget * **gwTableSofar**
- GtkWidget * **gwHRTime**
- GtkWidget * **gwVRLevel**

- GtkWidget * **gwDASofar**
- **clCfgFile CfgFile**
- **clGtkUtils GtkUtils**
- **clSockClie SClient**
- **clSockOp SOp**
- **clLevelMsg Msg**

7.46.1 Detailed Description

GUI for level server.

Definition at line 57 of file GUILevel.hh.

7.46.2 Constructor & Destructor Documentation

7.46.2.1 clGUILevel::clGUILevel ()

Definition at line 550 of file GUILevel.cc.

References bConnected, bRun, lResultCount, lResultPos, and spResults.

7.46.2.2 clGUILevel::~clGUILevel ()

Definition at line 560 of file GUILevel.cc.

References spResults.

7.46.3 Member Function Documentation

7.46.3.1 bool clGUILevel::GetCfg () `[private]`

Definition at line 104 of file GUILevel.cc.

References CfgFile, fDisplayHigh, fDisplayLow, _stLevelReq::fIntegrationTime, clCfgFile::GetFlt(), clCfgFile::GetInt(), GUILEV_CFGFILE, iBeamCount, clCfgFile::SetFileName(), and sRequest.

Referenced by Main().

7.46.3.2 bool clGUILevel::Build () `[private]`

Definition at line 116 of file GUILevel.cc.

References BuildDrawingPrims(), BuildSofar(), BuildTable1(), BuildTable2(), ConnectSignals(), cpWindowTxt, gcCrossHair, GUILEV_WSPACING, guSbCtxt, gwDASofar, gwStatusBar, gwVBox, and gwWindow.

Referenced by Main().

7.46.3.3 bool clGUILevel::BuildTable1 () [private]

Definition at line 144 of file GUILevel.cc.

References clGtkUtils::ComboListFromFile(), cpaLChannelTxt, cpBConnectTxt, cpLServerTxt, glServer, goAChannel, GtkUtils, GUILEV_CH_LOWER, GUILEV_CH_UPPER, GUILEV_HOSTFILE, GUILEV_SERVER_MAXLEN, GUILEV_WSPACING, gwBConnect, gwCServer, gwLChannel, gwLServer, gwSBChannel, gwTable1, gwVBox, and iBeamCount.

Referenced by Build().

7.46.3.4 bool clGUILevel::BuildTable2 () [private]

Definition at line 223 of file GUILevel.cc.

References clGtkUtils::BuildOptionMenu(), cpaLAlgorithmMenu, cpLAlgorithmTxt, cpLDisplayHighTxt, cpLDisplayLowTxt, cpLHighFrequencyTxt, cpLIntegrationTimeTxt, cpLLowFrequencyTxt, fDisplayHigh, fDisplayLow, _stLevelReq::fIntegrationTime, GtkUtils, GUILEV_ALGORITHM_ITEMS, GUILEV_ENTRY_WIDTH, GUILEV_WSPACING, gwaMIAlgorithm, gwEDisplayHigh, gwEDisplayLow, gwEHighFrequency, gwEIntegrationTime, gwELowFrequency, gwLAlgorithm, gwLDisplayHigh, gwLDisplayLow, gwLHighFrequency, gwLIntegrationTime, gwLLowFrequency, gwMAlgorithm, gwOMAlgorithm, gwTable2, gwVBox, and sRequest.

Referenced by Build().

7.46.3.5 bool clGUILevel::BuildSofar () [private]

Definition at line 358 of file GUILevel.cc.

References fDisplayHigh, fDisplayLow, GUILEV_SOFAR_HEIGHT, GUILEV_SOFAR_WIDTH, gwDASofar, gwHRTime, gwTableSofar, gwVBox, and gwVRLevel.

Referenced by Build().

7.46.3.6 bool clGUILevel::ConnectSignals () [private]

Definition at line 405 of file GUILevel.cc.

References clGtkUtils::ConnectMotionEvent(), GtkUtils, gwBConnect, gwDASofar, gwHRTime, gwVRLevel, gwWindow, WrapOnConnectClick(), WrapOnDelete(), WrapOnSofarConfigure(), WrapOnSofarExpose(), and WrapOnSofarMotion().

Referenced by Build().

7.46.3.7 bool clGUILevel::BuildDrawingPrims () [private]

Definition at line 427 of file GUILevel.cc.

References gcCrossHair, ggcSofarBG, ggcSofarFG, GUILEV_LINE_BG, GUILEV_LINE_FG, and gwDASofar.

Referenced by Build().

7.46.3.8 bool clGUILevel::ConnectToServer (const char *, int) [private]

Definition at line 447 of file GUILevel.cc.

References clSockClie::Connect(), GLOBAL_HEADER_LEN, GUILEV_REQ_-PROC, SClient, clSockOp::SetHandle(), SOp, and clSockOp::WriteN().

Referenced by OnConnectClick().

7.46.3.9 bool clGUILevel::SendSettings () [private]

Definition at line 462 of file GUILevel.cc.

References _stLevelReq::fHighFrequency, _stLevelReq::fIntegrationTime, _st-LevelReq::fLowFrequency, GLOBAL_HEADER_LEN, GtkUtils, GUILEV_-ALGORITHM_ITEMS, gwaMIAlgorithm, gwEHighFrequency, gwEIntegration-Time, gwELowFrequency, gwOMAlgorithm, gwSBChannel, _stLevelReq::i-Algorithm, iBeamCount, _stLevelReq::iChannel, Msg, clGtkUtils::OptionMenu-GetActive(), clLevelMsg::SetRequest(), SOp, sRequest, and clSockOp::Write-N().

Referenced by OnConnectClick().

7.46.3.10 void clGUILevel::DisplayResults () [private]

Definition at line 495 of file GUILevel.cc.

References fDisplayHigh, fDisplayLow, ggcSofarFG, gwDASofar, lResultCount, lResultPos, and spResults.

Referenced by OnGdkInput().

7.46.3.11 void clGUILevel::InitializeDisplay () [private]

Definition at line 529 of file GUILevel.cc.

References fDisplayHigh, fDisplayLow, gwDASofar, gwEDisplayHigh, gw-EDisplayLow, gwEIntegrationTime, gwHRTime, gwVRLevel, lResultPos, and spResults.

Referenced by OnGdkInput().

7.46.3.12 int clGUILevel::Main (int ∗, char ∗∗∗)

Definition at line 566 of file GUILevel.cc.

References Build(), cpWindowTxt, GetCfg(), GUILEV_VER_MAJ, GUILEV_VER_MIN, and GUILEV_VER_PL.

Referenced by main().

7.46.3.13 gboolean clGUILevel::OnDelete (GtkWidget ∗, GdkEvent ∗, gpointer)

Definition at line 585 of file GUILevel.cc.

References bRun.

Referenced by WrapOnDelete().

7.46.3.14 void clGUILevel::OnConnectClick (GtkButton ∗, gpointer)

Definition at line 594 of file GUILevel.cc.

References bConnected, bFirstResult, clSockOp::Close(), ConnectToServer(), clSockOp::GetHandle(), gwCServer, iGdkInputTag, SendSettings(), SOp, and WrapOnGdkInput().

Referenced by WrapOnConnectClick().

7.46.3.15 gboolean clGUILevel::OnSofarExpose (GtkWidget ∗, GdkEventExpose ∗, gpointer)

Definition at line 640 of file GUILevel.cc.

References fDisplayHigh, fDisplayLow, ggcSofarFG, lResultPos, and spResults.

Referenced by WrapOnSofarExpose().

7.46.3.16 gboolean clGUILevel::OnSofarMotion (GtkWidget ∗, GdkEventMotion ∗, gpointer)

Definition at line 664 of file GUILevel.cc.

References fDisplayHigh, fDisplayLow, guSbCtxt, gwStatusBar, lResultPos, and spResults.

Referenced by WrapOnSofarMotion().

7.46.3.17 gboolean clGUILevel::OnSofarConfigure (GtkWidget ∗, GdkEventConfigure ∗, gpointer)

Definition at line 685 of file GUILevel.cc.

References lResultCount, and spResults.

Referenced by WrapOnSofarConfigure().

7.46.3.18 void clGUILevel::OnGdkInput (gpointer, gint, GdkInputCondition)

Definition at line 697 of file GUILevel.cc.

References bConnected, bFirstResult, bRun, clSockOp::Close(), cpResultBuf, DisplayResults(), clLevelMsg::GetResult(), GLOBAL_HEADER_LEN, iGdkInputTag, InitializeDisplay(), lResultCount, lResultPos, Msg, clSockOp::ReadN(), clSockOp::ReadSelect(), SOp, and spResults.

Referenced by WrapOnGdkInput().

7.46.4 Member Data Documentation

7.46.4.1 bool clGUILevel::bRun `[private]`

Definition at line 59 of file GUILevel.hh.

Referenced by clGUILevel(), OnDelete(), and OnGdkInput().

7.46.4.2 bool clGUILevel::bConnected `[private]`

Definition at line 60 of file GUILevel.hh.

Referenced by clGUILevel(), OnConnectClick(), and OnGdkInput().

7.46.4.3 bool clGUILevel::bFirstResult `[private]`

Definition at line 61 of file GUILevel.hh.

Referenced by OnConnectClick(), and OnGdkInput().

7.46.4.4 int clGUILevel::iBeamCount `[private]`

Definition at line 62 of file GUILevel.hh.

Referenced by BuildTable1(), GetCfg(), and SendSettings().

7.46.4.5 long clGUILevel::lResultPos `[private]`

Definition at line 63 of file GUILevel.hh.

Referenced by clGUILevel(), DisplayResults(), InitializeDisplay(), OnGdkInput(), OnSofarExpose(), and OnSofarMotion().

7.46.4.6 long clGUILevel::lResultCount [private]

Definition at line 64 of file GUILevel.hh.

Referenced by clGUILevel(), DisplayResults(), OnGdkInput(), and OnSofarConfigure().

7.46.4.7 float clGUILevel::fDisplayLow [private]

Definition at line 65 of file GUILevel.hh.

Referenced by BuildSofar(), BuildTable2(), DisplayResults(), GetCfg(), InitializeDisplay(), OnSofarExpose(), and OnSofarMotion().

7.46.4.8 float clGUILevel::fDisplayHigh [private]

Definition at line 66 of file GUILevel.hh.

Referenced by BuildSofar(), BuildTable2(), DisplayResults(), GetCfg(), InitializeDisplay(), OnSofarExpose(), and OnSofarMotion().

7.46.4.9 char clGUILevel::cpResultBuf[GLOBAL_HEADER_LEN] [private]

Definition at line 67 of file GUILevel.hh.

Referenced by OnGdkInput().

7.46.4.10 stLevelReq clGUILevel::sRequest [private]

Definition at line 68 of file GUILevel.hh.

Referenced by BuildTable2(), GetCfg(), and SendSettings().

7.46.4.11 stpLevelRes clGUILevel::spResults [private]

Definition at line 69 of file GUILevel.hh.

Referenced by clGUILevel(), DisplayResults(), InitializeDisplay(), OnGdkInput(), OnSofarConfigure(), OnSofarExpose(), OnSofarMotion(), and ~clGUILevel().

7.46.4.12 guint clGUILevel::guSbCtxt [private]

Definition at line 71 of file GUILevel.hh.

Referenced by Build(), and OnSofarMotion().

7.46.4.13 gint clGUILevel::iGdkInputTag [private]

Definition at line 72 of file GUILevel.hh.

Referenced by OnConnectClick(), and OnGdkInput().

7.46.4.14 GList∗ clGUILevel::glServer [private]

Definition at line 73 of file GUILevel.hh.

Referenced by BuildTable1().

7.46.4.15 GdkGC∗ clGUILevel::ggcSofarBG [private]

Definition at line 76 of file GUILevel.hh.

Referenced by BuildDrawingPrims().

7.46.4.16 GdkGC∗ clGUILevel::ggcSofarFG [private]

Definition at line 77 of file GUILevel.hh.

Referenced by BuildDrawingPrims(), DisplayResults(), and OnSofarExpose().

7.46.4.17 GdkCursor∗ clGUILevel::gcCrossHair [private]

Definition at line 78 of file GUILevel.hh.

Referenced by Build(), and BuildDrawingPrims().

7.46.4.18 GtkWidget∗ clGUILevel::gwWindow [private]

Definition at line 82 of file GUILevel.hh.

Referenced by Build(), and ConnectSignals().

7.46.4.19 GtkWidget∗ clGUILevel::gwVBox [private]

Definition at line 83 of file GUILevel.hh.

Referenced by Build(), BuildSofar(), BuildTable1(), and BuildTable2().

7.46.4.20 GtkWidget∗ clGUILevel::gwStatusBar [private]

Definition at line 84 of file GUILevel.hh.

Referenced by Build(), and OnSofarMotion().

7.46.4.21 GtkWidget* clGUILevel::gwTable1 [private]

Definition at line 86 of file GUILevel.hh.

Referenced by BuildTable1().

7.46.4.22 GtkWidget* clGUILevel::gwLServer [private]

Definition at line 87 of file GUILevel.hh.

Referenced by BuildTable1().

7.46.4.23 GtkWidget* clGUILevel::gwCServer [private]

Definition at line 88 of file GUILevel.hh.

Referenced by BuildTable1(), and OnConnectClick().

7.46.4.24 GtkWidget* clGUILevel::gwLChannel [private]

Definition at line 89 of file GUILevel.hh.

Referenced by BuildTable1().

7.46.4.25 GtkObject* clGUILevel::goAChannel [private]

Definition at line 90 of file GUILevel.hh.

Referenced by BuildTable1().

7.46.4.26 GtkWidget* clGUILevel::gwSBChannel [private]

Definition at line 91 of file GUILevel.hh.

Referenced by BuildTable1(), and SendSettings().

7.46.4.27 GtkWidget* clGUILevel::gwBConnect [private]

Definition at line 92 of file GUILevel.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.46.4.28 GtkWidget* clGUILevel::gwTable2 [private]

Definition at line 94 of file GUILevel.hh.

Referenced by BuildTable2().

7.46.4.29 GtkWidget* clGUILevel::gwLAlgorithm [private]

Definition at line 95 of file GUILevel.hh.

Referenced by BuildTable2().

7.46.4.30 GtkWidget* clGUILevel::gwOMAlgorithm [private]

Definition at line 96 of file GUILevel.hh.

Referenced by BuildTable2(), and SendSettings().

7.46.4.31 GtkWidget* clGUILevel::gwMAlgorithm [private]

Definition at line 97 of file GUILevel.hh.

Referenced by BuildTable2().

7.46.4.32 GtkWidget* clGUILevel::gwaMIAlgorithm[GUILEV_-ALGORITHM_ITEMS] [private]

Definition at line 98 of file GUILevel.hh.

Referenced by BuildTable2(), and SendSettings().

7.46.4.33 GtkWidget* clGUILevel::gwLIntegrationTime [private]

Definition at line 99 of file GUILevel.hh.

Referenced by BuildTable2().

7.46.4.34 GtkWidget* clGUILevel::gwEIntegrationTime [private]

Definition at line 100 of file GUILevel.hh.

Referenced by BuildTable2(), InitializeDisplay(), and SendSettings().

7.46.4.35 GtkWidget* clGUILevel::gwLLowFrequency [private]

Definition at line 101 of file GUILevel.hh.

Referenced by BuildTable2().

7.46.4.36 GtkWidget* clGUILevel::gwELowFrequency [private]

Definition at line 102 of file GUILevel.hh.

Referenced by BuildTable2(), and SendSettings().

7.46.4.37 GtkWidget* clGUILevel::gwLHighFrequency [private]

Definition at line 103 of file GUILevel.hh.

Referenced by BuildTable2().

7.46.4.38 GtkWidget* clGUILevel::gwEHighFrequency [private]

Definition at line 104 of file GUILevel.hh.

Referenced by BuildTable2(), and SendSettings().

7.46.4.39 GtkWidget* clGUILevel::gwLDisplayLow [private]

Definition at line 105 of file GUILevel.hh.

Referenced by BuildTable2().

7.46.4.40 GtkWidget* clGUILevel::gwEDisplayLow [private]

Definition at line 106 of file GUILevel.hh.

Referenced by BuildTable2(), and InitializeDisplay().

7.46.4.41 GtkWidget* clGUILevel::gwLDisplayHigh [private]

Definition at line 107 of file GUILevel.hh.

Referenced by BuildTable2().

7.46.4.42 GtkWidget* clGUILevel::gwEDisplayHigh [private]

Definition at line 108 of file GUILevel.hh.

Referenced by BuildTable2(), and InitializeDisplay().

7.46.4.43 GtkWidget* clGUILevel::gwTableSofar [private]

Definition at line 110 of file GUILevel.hh.

Referenced by BuildSofar().

7.46.4.44 GtkWidget* clGUILevel::gwHRTime [private]

Definition at line 111 of file GUILevel.hh.

Referenced by BuildSofar(), ConnectSignals(), and InitializeDisplay().

7.46.4.45 GtkWidget* clGUILevel::gwVRLevel [private]

Definition at line 112 of file GUILevel.hh.

Referenced by BuildSofar(), ConnectSignals(), and InitializeDisplay().

7.46.4.46 GtkWidget* clGUILevel::gwDASofar [private]

Definition at line 113 of file GUILevel.hh.

Referenced by Build(), BuildDrawingPrims(), BuildSofar(), ConnectSignals(), DisplayResults(), and InitializeDisplay().

7.46.4.47 clCfgFile clGUILevel::CfgFile [private]

Definition at line 116 of file GUILevel.hh.

Referenced by GetCfg().

7.46.4.48 clGtkUtils clGUILevel::GtkUtils [private]

Definition at line 117 of file GUILevel.hh.

Referenced by BuildTable1(), BuildTable2(), ConnectSignals(), and SendSettings().

7.46.4.49 clSockClie clGUILevel::SClient [private]

Definition at line 118 of file GUILevel.hh.

Referenced by ConnectToServer().

7.46.4.50 clSockOp clGUILevel::SOp [private]

Definition at line 119 of file GUILevel.hh.

Referenced by ConnectToServer(), OnConnectClick(), OnGdkInput(), and SendSettings().

7.46.4.51 clLevelMsg clGUILevel::Msg [private]

Definition at line 120 of file GUILevel.hh.

Referenced by OnGdkInput(), and SendSettings().

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

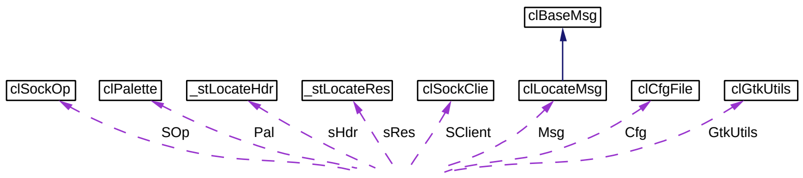
- **GUILevel.hh**
- **GUILevel.cc**

7.47 clGUILocate Class Reference

GUI for locating.

`#include <GUILocate.hh>`

Collaboration diagram for clGUILocate:



Public Member Functions

- **clGUILocate** ()
- ~**clGUILocate** ()
- int **Main** (int ∗, char ∗∗∗)
- gboolean **OnDelete** (GtkWidget ∗, GdkEvent ∗, gpointer)
- void **OnConnectClick** (GtkButton ∗, gpointer)
- void **OnPaletteActivate** (GtkMenuItem ∗, gpointer)
- gboolean **OnLocateExpose** (GtkWidget ∗, GdkEventExpose ∗, gpointer)
- gboolean **OnLocateMotion** (GtkWidget ∗, GdkEventMotion ∗, gpointer)
- void **OnGdkInput** (gpointer, gint, GdkInputCondition)

Private Member Functions

- bool **GetCfg** ()
- bool **Build** ()
- bool **BuildTable1** ()
- bool **BuildTable2** ()
- bool **BuildLocate** ()
- bool **ConnectSignals** ()
- bool **BuildDrawingPrims** ()
- bool **ConnectToServer** (const char ∗, int)
- void **DisplayResults** ()

Private Attributes

- bool **bRun**
- bool **bConnected**
- int **iPalette**

- int **iMsgSize**
- **stLocateHdr sHdr**
- **stLocateRes sRes**
- gint **iGdkInputTag**
- GList * **glServer**
- GdkGC * **ggcLocateBG**
- GdkGC * **ggcLocateFG**
- GdkCursor * **gcCrossHair**
- GtkWidget * **gwWindow**
- GtkWidget * **gwVBox**
- GtkWidget * **gwTable1**
- GtkWidget * **gwLServer**
- GtkWidget * **gwCServer**
- GtkWidget * **gwBConnect**
- GtkWidget * **gwTable2**
- GtkWidget * **gwLPalette**
- GtkWidget * **gwOMPalette**
- GtkWidget * **gwMPalette**
- GtkWidget * **gwaMIPalette** [GUILOC_PALETTE_ITEMS]
- GtkWidget * **gwSWLocate**
- GtkWidget * **gwDALocate**
- clAlloc **ResMsg**
- clAlloc **ResMatrix**
- clAlloc **ResFrame**
- **clCfgFile Cfg**
- **clGtkUtils GtkUtils**
- **clLocateMsg Msg**
- **clPalette Pal**
- **clSockClie SClient**
- **clSockOp SOp**

7.47.1 Detailed Description

GUI for locating.

Definition at line 71 of file GUILocate.hh.

7.47.2 Constructor & Destructor Documentation

7.47.2.1 clGUILocate::clGUILocate ()

Definition at line 345 of file GUILocate.cc.

References bConnected, and bRun.

7.47.2.2 clGUILocate::~clGUILocate ()

Definition at line 352 of file GUILocate.cc.

7.47.3 Member Function Documentation

7.47.3.1 bool clGUILocate::GetCfg () [private]

Definition at line 98 of file GUILocate.cc.

References Cfg, clPalette::GenBW(), clCfgFile::GetInt(), GUILOC_CFGFILE, iPalette, OnPaletteActivate(), Pal, and clCfgFile::SetFileName().

Referenced by Main().

7.47.3.2 bool clGUILocate::Build () [private]

Definition at line 114 of file GUILocate.cc.

References BuildDrawingPrims(), BuildLocate(), BuildTable1(), BuildTable2(), ConnectSignals(), cpWindowTxt, gcCrossHair, GUILOC_WSPACING, gwDALocate, gwVBox, and gwWindow.

Referenced by Main().

7.47.3.3 bool clGUILocate::BuildTable1 () [private]

Definition at line 135 of file GUILocate.cc.

References clGtkUtils::ComboListFromFile(), cpBConnectTxt, cpLServerTxt, glServer, GtkUtils, GUILOC_HOSTFILE, GUILOC_SERVER_MAXLEN, GUILOC_WSPACING, gwBConnect, gwCServer, gwLServer, gwTable1, and gwVBox.

Referenced by Build().

7.47.3.4 bool clGUILocate::BuildTable2 () [private]

Definition at line 179 of file GUILocate.cc.

References clGtkUtils::BuildOptionMenu(), cpaLPaletteMenu, cpLPaletteTxt, GtkUtils, GUILOC_PALETTE_ITEMS, GUILOC_WSPACING, gwaMIPalette, gwLPalette, gwMPalette, gwOMPalette, gwTable2, gwVBox, and iPalette.

Referenced by Build().

7.47.3.5 bool clGUILocate::BuildLocate () [private]

Definition at line 210 of file GUILocate.cc.

References gwDALocate, gwSWLocate, and gwVBox.

Referenced by Build().

7.47.3.6 bool clGUILocate::ConnectSignals () [private]

Definition at line 229 of file GUILocate.cc.

References GUILOC_PALETTE_ITEMS, gwaMIPalette, gwBConnect, gwDALocate, gwWindow, WrapOnConnectClick(), WrapOnDelete(), WrapOnLocateExpose(), WrapOnLocateMotion(), and WrapOnPaletteActivate().

Referenced by Build().

7.47.3.7 bool clGUILocate::BuildDrawingPrims () [private]

Definition at line 256 of file GUILocate.cc.

References gcCrossHair, ggcLocateBG, ggcLocateFG, GUILOC_LOCATE_BG, GUILOC_LOCATE_FG, and gwDALocate.

Referenced by Build().

7.47.3.8 bool clGUILocate::ConnectToServer (const char *, int) [private]

Definition at line 276 of file GUILocate.cc.

References clSockClie::Connect(), clLocateMsg::GetHeader(), GLOBAL_HEADER_LEN, Msg, clSockOp::ReadN(), SClient, clSockOp::SetHandle(), sHdr, and SOp.

Referenced by OnConnectClick().

7.47.3.9 void clGUILocate::DisplayResults () [private]

Definition at line 294 of file GUILocate.cc.

References GDT, ggcLocateFG, gwDALocate, gwSWLocate, _stLocateHdr::iHeight, _stLocateHdr::iWidth, _stLocateRes::lPointCount, Pal, ResFrame, ResMatrix, sHdr, clPalette::Size(), and sRes.

Referenced by OnGdkInput().

7.47.3.10 int clGUILocate::Main (int *, char ***)

Definition at line 357 of file GUILocate.cc.

References Build(), cpWindowTxt, GetCfg(), GUILOC_VER_MAJ, GUILOC_VER_MIN, and GUILOC_VER_PL.

Referenced by main().

7.47.3.11 gboolean clGUILocate::OnDelete (GtkWidget *, GdkEvent *, gpointer)

Definition at line 376 of file GUILocate.cc.

References bRun.

Referenced by WrapOnDelete().

7.47.3.12 void clGUILocate::OnConnectClick (GtkButton *, gpointer)

Definition at line 385 of file GUILocate.cc.

References bConnected, clSockOp::Close(), ConnectToServer(), GDT, clSockOp::GetHandle(), GLOBAL_HEADER_LEN, gwCServer, gwDALocate, iGdkInputTag, _stLocateHdr::iHeight, iMsgSize, _stLocateHdr::iWidth, ResFrame, ResMatrix, ResMsg, sHdr, SOp, and WrapOnGdkInput().

Referenced by WrapOnConnectClick().

7.47.3.13 void clGUILocate::OnPaletteActivate (GtkMenuItem *, gpointer)

Definition at line 429 of file GUILocate.cc.

References clPalette::GenBW(), clPalette::GenDir(), clPalette::GenGreen(), clPalette::GenGreen2(), clPalette::GenHSV(), clPalette::GenLight(), clPalette::GenPureGreen(), clPalette::GenTemp(), clPalette::GenWB(), GUILOC_PAL_BW, GUILOC_PAL_DIR, GUILOC_PAL_GREEN, GUILOC_PAL_GREEN2, GUILOC_PAL_HSV, GUILOC_PAL_LIGHT, GUILOC_PAL_PUREGREEN, GUILOC_PAL_TEMP, GUILOC_PAL_WB, iPalette, and Pal.

Referenced by GetCfg(), and WrapOnPaletteActivate().

7.47.3.14 gboolean clGUILocate::OnLocateExpose (GtkWidget *, GdkEventExpose *, gpointer)

Definition at line 467 of file GUILocate.cc.

References ggcLocateFG, gwDALocate, _stLocateHdr::iHeight, _stLocateHdr::iWidth, ResFrame, and sHdr.

Referenced by WrapOnLocateExpose().

7.47.3.15 gboolean clGUILocate::OnLocateMotion (GtkWidget *, GdkEventMotion *, gpointer)

Definition at line 497 of file GUILocate.cc.

Referenced by WrapOnLocateMotion().

7.47.3.16 void clGUILocate::OnGdkInput (gpointer, gint, GdkInputCondition)

Definition at line 504 of file GUILocate.cc.

References bConnected, bRun, clSockOp::Close(), DisplayResults(), GDT, clLocateMsg::GetResult(), iGdkInputTag, iMsgSize, Msg, clSockOp::ReadN(), clSockOp::ReadSelect(), ResMatrix, ResMsg, SOp, and sRes.

Referenced by WrapOnGdkInput().

7.47.4 Member Data Documentation

7.47.4.1 bool clGUILocate::bRun [private]

Definition at line 73 of file GUILocate.hh.

Referenced by clGUILocate(), OnDelete(), and OnGdkInput().

7.47.4.2 bool clGUILocate::bConnected [private]

Definition at line 74 of file GUILocate.hh.

Referenced by clGUILocate(), OnConnectClick(), and OnGdkInput().

7.47.4.3 int clGUILocate::iPalette [private]

Definition at line 75 of file GUILocate.hh.

Referenced by BuildTable2(), GetCfg(), and OnPaletteActivate().

7.47.4.4 int clGUILocate::iMsgSize [private]

Definition at line 76 of file GUILocate.hh.

Referenced by OnConnectClick(), and OnGdkInput().

7.47.4.5 stLocateHdr clGUILocate::sHdr [private]

Definition at line 77 of file GUILocate.hh.

Referenced by ConnectToServer(), DisplayResults(), OnConnectClick(), and OnLocateExpose().

7.47.4.6 stLocateRes clGUILocate::sRes [private]

Definition at line 78 of file GUILocate.hh.

Referenced by DisplayResults(), and OnGdkInput().

7.47.4.7 gint clGUILocate::iGdkInputTag [private]

Definition at line 80 of file GUILocate.hh.

Referenced by OnConnectClick(), and OnGdkInput().

7.47.4.8 GList* clGUILocate::glServer [private]

Definition at line 81 of file GUILocate.hh.

Referenced by BuildTable1().

7.47.4.9 GdkGC* clGUILocate::ggcLocateBG [private]

Definition at line 84 of file GUILocate.hh.

Referenced by BuildDrawingPrims().

7.47.4.10 GdkGC* clGUILocate::ggcLocateFG [private]

Definition at line 85 of file GUILocate.hh.

Referenced by BuildDrawingPrims(), DisplayResults(), and OnLocateExpose().

7.47.4.11 GdkCursor* clGUILocate::gcCrossHair [private]

Definition at line 86 of file GUILocate.hh.

Referenced by Build(), and BuildDrawingPrims().

7.47.4.12 GtkWidget* clGUILocate::gwWindow [private]

Definition at line 90 of file GUILocate.hh.

Referenced by Build(), and ConnectSignals().

7.47.4.13 GtkWidget* clGUILocate::gwVBox [private]

Definition at line 91 of file GUILocate.hh.

Referenced by Build(), BuildLocate(), BuildTable1(), and BuildTable2().

7.47.4.14 GtkWidget* clGUILocate::gwTable1 [private]

Definition at line 93 of file GUILocate.hh.

Referenced by BuildTable1().

7.47.4.15 GtkWidget* clGUILocate::gwLServer [private]

Definition at line 94 of file GUILocate.hh.

Referenced by BuildTable1().

7.47.4.16 GtkWidget* clGUILocate::gwCServer [private]

Definition at line 95 of file GUILocate.hh.

Referenced by BuildTable1(), and OnConnectClick().

7.47.4.17 GtkWidget* clGUILocate::gwBConnect [private]

Definition at line 96 of file GUILocate.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.47.4.18 GtkWidget* clGUILocate::gwTable2 [private]

Definition at line 98 of file GUILocate.hh.

Referenced by BuildTable2().

7.47.4.19 GtkWidget* clGUILocate::gwLPalette [private]

Definition at line 99 of file GUILocate.hh.

Referenced by BuildTable2().

7.47.4.20 GtkWidget* clGUILocate::gwOMPalette [private]

Definition at line 100 of file GUILocate.hh.

Referenced by BuildTable2().

7.47.4.21 GtkWidget* clGUILocate::gwMPalette [private]

Definition at line 101 of file GUILocate.hh.

Referenced by BuildTable2().

7.47.4.22 GtkWidget* clGUILocate::gwaMIPalette[GUILOC_PALETTE_ITEMS] [private]

Definition at line 102 of file GUILocate.hh.

Referenced by BuildTable2(), and ConnectSignals().

7.47.4.23 GtkWidget* clGUILocate::gwSWLocate [private]

Definition at line 104 of file GUILocate.hh.

Referenced by BuildLocate(), and DisplayResults().

7.47.4.24 GtkWidget* clGUILocate::gwDALocate [private]

Definition at line 105 of file GUILocate.hh.

Referenced by Build(), BuildDrawingPrims(), BuildLocate(), ConnectSignals(), DisplayResults(), OnConnectClick(), and OnLocateExpose().

7.47.4.25 clAlloc clGUILocate::ResMsg [private]

Definition at line 107 of file GUILocate.hh.

Referenced by OnConnectClick(), and OnGdkInput().

7.47.4.26 clAlloc clGUILocate::ResMatrix [private]

Definition at line 108 of file GUILocate.hh.

Referenced by DisplayResults(), OnConnectClick(), and OnGdkInput().

7.47.4.27 clAlloc clGUILocate::ResFrame [private]

Definition at line 109 of file GUILocate.hh.

Referenced by DisplayResults(), OnConnectClick(), and OnLocateExpose().

7.47.4.28 clCfgFile clGUILocate::Cfg [private]

Definition at line 110 of file GUILocate.hh.

Referenced by GetCfg().

7.47.4.29 clGtkUtils clGUILocate::GtkUtils [private]

Definition at line 111 of file GUILocate.hh.

Referenced by BuildTable1(), and BuildTable2().

7.47.4.30 clLocateMsg clGUILocate::Msg [private]

Definition at line 112 of file GUILocate.hh.

Referenced by ConnectToServer(), and OnGdkInput().

7.47.4.31 clPalette clGUILocate::Pal [private]

Definition at line 113 of file GUILocate.hh.

Referenced by DisplayResults(), GetCfg(), and OnPaletteActivate().

7.47.4.32 clSockClie clGUILocate::SClient [private]

Definition at line 114 of file GUILocate.hh.

Referenced by ConnectToServer().

7.47.4.33 clSockOp clGUILocate::SOp [private]

Definition at line 115 of file GUILocate.hh.

Referenced by ConnectToServer(), OnConnectClick(), and OnGdkInput().

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

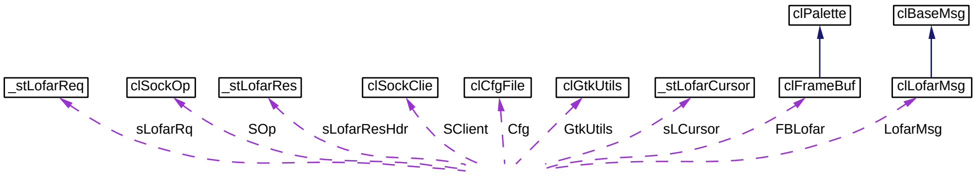
- **GUILocate.hh**
- **GUILocate.cc**

7.48 clGUILofar Class Reference

LOFAR/DEMON GUI.

`#include <GUILofar.hh>`

Collaboration diagram for clGUILofar:



Public Member Functions

- **clGUILofar** (int ∗, char ∗∗∗)
- ∼**clGUILofar** ()
- int **Exec** ()
- gint **OnDelete** (GtkWidget ∗, GdkEventAny ∗)
- void **OnHideToggled** (GtkToggleButton ∗, gpointer)
- gint **OnConnectClick** (GtkWidget ∗, gpointer)
- void **OnFreezeToggled** (GtkToggleButton ∗, gpointer)
- gint **OnExposeLofar** (GtkWidget ∗, GdkEventExpose ∗, gpointer)
- gint **OnConfigureLofar** (GtkWidget ∗, GdkEventConfigure ∗, gpointer)
- gint **OnExposeLine** (GtkWidget ∗, GdkEventExpose ∗, gpointer)
- gint **OnExposeCursor** (GtkWidget ∗, GdkEventExpose ∗, gpointer)
- gint **OnMotionLofar** (GtkWidget ∗, GdkEventMotion ∗, gpointer)
- gint **OnPaletteActivate** (GtkWidget ∗, gpointer)
- void **OnAverageToggled** (GtkToggleButton ∗, gpointer)
- void **OnClipValueChanged** (GtkAdjustment ∗, gpointer)
- void **OnSaveClicks** (GtkWidget ∗, gpointer)
- void **OnGdkInput** (gpointer, gint, GdkInputCondition)

Private Member Functions

- bool **Build** ()
- bool **BuildTable1** ()
- bool **BuildTable2** ()
- bool **BuildTable3** ()
- bool **BuildTable4** ()
- bool **BuildTableLofar** ()
- bool **BuildDrawingPrims** ()
- void **FreeDrawingPrims** ()
- bool **ConnectSignals** ()

- bool **ParseServerStr** (char ∗, int ∗, const char ∗)
- bool **InitConnection** (const char ∗, int)
- bool **SendSettings** ()
- void **PrintStatus** ()
- void **SetPalette** (int)
- void **Configure** ()
- void **ConfigureHeight** ()
- void **DrawCursor** ()
- void **SaveInfo** (const char ∗, time_t)
- void **StartNewImgFile** ()

Private Attributes

- bool **bRun**
- bool **bConnected**
- bool **bFreezed**
- bool **bConfigured**
- bool **bAveraged**
- bool **bCursorDrag**
- bool **bSaving**
- int **iResMsgBufSize**
- int **iFit**
- int **iPalette**
- int **iLofarWidth**
- int **iLofarHeight**
- int **iCursorX**
- int **iCursorY**
- int **iClips**
- int **iTIFFCompression**
- int **iJPEGQuality**
- int **iContSaveScans**
- int **iCompressMode**
- int **iScanCount**
- int **iImgCount**
- int **iBeamCount**
- long **lSpectSize**
- GDT **fClip**
- **stLofarReq sLofarRq**
- **stLofarRes sLofarResHdr**
- **stLofarCursor sLCursor**
- GList ∗ **glServer**
- gint **giGdkTag**
- guint **guSbCtxt**
- GtkWidget ∗ **gwWindow**
- GtkWidget ∗ **gwVBox**
- GtkWidget ∗ **gwCBHide**

- GtkWidget ∗ **gwStatusBar**
- GtkWidget ∗ **gwTable1**
- GtkWidget ∗ **gwLServer**
- GtkWidget ∗ **gwCServer**
- GtkWidget ∗ **gwLChannel**
- GtkObject ∗ **goAChannel**
- GtkWidget ∗ **gwSBChannel**
- GtkWidget ∗ **gwBConnect**
- GtkWidget ∗ **gwBDisconnect**
- GtkWidget ∗ **gwCBFreeze**
- GtkWidget ∗ **gwTable2**
- GtkWidget ∗ **gwLType**
- GtkWidget ∗ **gwOMType**
- GtkWidget ∗ **gwMType**
- GtkWidget ∗ **gwaMIType** [LGUI_TYPE_ITEMS]
- GtkWidget ∗ **gwLWindow**
- GtkWidget ∗ **gwOMWindow**
- GtkWidget ∗ **gwMWindow**
- GtkWidget ∗ **gwaMIWindow** [LGUI_WINDOW_ITEMS]
- GtkWidget ∗ **gwLWinParam**
- GtkWidget ∗ **gwEWinParam**
- GtkWidget ∗ **gwLWinLength**
- GtkWidget ∗ **gwOMWinLength**
- GtkWidget ∗ **gwMWinLength**
- GtkWidget ∗ **gwaMIWinLength** [LGUI_WIN_LENGTH_ITEMS]
- GtkWidget ∗ **gwLLowerFreq**
- GtkWidget ∗ **gwELowerFreq**
- GtkWidget ∗ **gwLHigherFreq**
- GtkWidget ∗ **gwEHigherFreq**
- GtkWidget ∗ **gwLOverlap**
- GtkWidget ∗ **gwEOverlap**
- GtkWidget ∗ **gwTable3**
- GtkWidget ∗ **gwLRemoveNoise**
- GtkWidget ∗ **gwOMRemoveNoise**
- GtkWidget ∗ **gwMRemoveNoise**
- GtkWidget ∗ **gwaMIRemoveNoise** [LGUI_REMOVE_NOISE_ITEMS]
- GtkWidget ∗ **gwLAlpha**
- GtkWidget ∗ **gwEAlpha**
- GtkWidget ∗ **gwLMeanLength**
- GtkWidget ∗ **gwEMeanLength**
- GtkWidget ∗ **gwLGapLength**
- GtkWidget ∗ **gwEGapLength**
- GtkWidget ∗ **gwLAverageCount**
- GtkWidget ∗ **gwEAverageCount**
- GtkWidget ∗ **gwLClip**
- GtkWidget ∗ **gwOMClip**

- GtkWidget * **gwMClip**
- GtkWidget * **gwaMIClip** [LGUI_CLIP_ITEMS]
- GtkWidget * **gwCBLinear**
- GtkWidget * **gwCBDemon**
- GtkWidget * **gwLPalette**
- GtkWidget * **gwOMPalette**
- GtkWidget * **gwMPalette**
- GtkWidget * **gwaMIPalette** [LGUI_PALETTE_ITEMS]
- GtkWidget * **gwCBAverage**
- GtkWidget * **gwCBSaving**
- GtkWidget * **gwBSave**
- GtkWidget * **gwFSSave**
- GtkWidget * **gwTable4**
- GtkWidget * **gwLClipValue**
- GtkObject * **goAClipValue**
- GtkWidget * **gwHSClipValue**
- GtkWidget * **gwTableLofar**
- GtkWidget * **gwLTopTime**
- GtkWidget * **gwLBottomTime**
- GtkWidget * **gwSWLofar**
- GtkWidget * **gwTableLofar2**
- GtkWidget * **gwHRFrequency**
- GtkWidget * **gwVRTime**
- GtkWidget * **gwDALine**
- GtkWidget * **gwDACursor**
- GtkWidget * **gwDALofar**
- GdkGC * **ggcLofarBG**
- GdkGC * **ggcLofarFG**
- GdkGC * **ggcLineBG**
- GdkGC * **ggcLineFG**
- GdkGC * **ggcCursorBG**
- GdkGC * **ggcCursorFG**
- GdkCursor * **gcCrossHair**
- std::string **strImgFileName**
- clAlloc **SpectData**
- clAlloc **AvgSpectData**
- **clCfgFile Cfg**
- clDSPOp **DSP**
- **clFrameBuf FBLofar**
- **clGtkUtils GtkUtils**
- **clLofarMsg LofarMsg**
- **clSockClie SClient**
- **clSockOp SOp**

7.48.1 Detailed Description

LOFAR/DEMON GUI.

Definition at line 119 of file GUILofar.hh.

7.48.2 Constructor & Destructor Documentation

7.48.2.1 clGUILofar::clGUILofar (int *, char ***)

Definition at line 1331 of file GUILofar.cc.

References bAveraged, bConfigured, bConnected, bCursorDrag, bFreezed, bRun, Cfg, cpWindowTxt, _stLofarReq::fAlpha, FB_TIFF_COMPRESS_DEFLATE, FB_TIFF_COMPRESS_JPEG, FB_TIFF_COMPRESS_LZW, FB_TIFF_COMPRESS_NONE, FB_TIFF_COMPRESS_RLE, fClip, _stLofarReq::fHighFreq, _stLofarReq::fLowFreq, clCfgFile::GetFlt(), clCfgFile::GetInt(), glServer, iBeamCount, _stLofarReq::iClip, iCompressMode, iContSaveScans, iFit, iJPEGQuality, iLofarHeight, iLofarWidth, iPalette, _stLofarReq::iRemoveNoise, iTIFFCompression, _stLofarReq::lAvgCount, _stLofarReq::lGapLength, LGUI_CFGFILE, LGUI_DEF_ALPHA, LGUI_DEF_GAP_LENGTH, LGUI_DEF_HEIGHT, LGUI_DEF_HIGH_FREQ, LGUI_DEF_LOW_FREQ, LGUI_DEF_MEAN_LENGTH, LGUI_DEF_REMOVE_NOISE, LGUI_DEF_WIN_LENGTH, LGUI_VER_MAJ, LGUI_VER_MIN, LGUI_VER_PL, _stLofarReq::lMeanLength, _stLofarReq::lWinLength, MSG_LOFAR_CLIP_NONE, clCfgFile::SetFileName(), sLCursor, and sLofarRq.

7.48.2.2 clGUILofar::~clGUILofar ()

Definition at line 1410 of file GUILofar.cc.

7.48.3 Member Function Documentation

7.48.3.1 bool clGUILofar::Build () [private]

Definition at line 189 of file GUILofar.cc.

References BuildDrawingPrims(), BuildTable1(), BuildTable2(), BuildTable3(), BuildTable4(), BuildTableLofar(), cpWindowTxt, gcCrossHair, guSbCtxt, gwCBHide, gwDALine, gwDALofar, gwStatusBar, gwVBox, gwWindow, and LGUI_PADDING.

Referenced by Exec().

7.48.3.2 bool clGUILofar::BuildTable1 () [private]

Definition at line 263 of file GUILofar.cc.

References clGtkUtils::ComboListFromFile(), cpaLChannelTxt, cpBConnectTxt, cpBDisconnectTxt, cpCBFreezeTxt, cpLServerTxt, glServer, goAChannel, GtkUtils, gwBConnect, gwBDisconnect, gwCBFreeze, gwCServer, gwLChannel, gwLServer, gwSBChannel, gwTable1, gwVBox, iBeamCount, LGUI_CH_LOWER, LGUI_CH_UPPER, LGUI_HOSTFILE, LGUI_PADDING, and LGUI_SERVER_MAXLEN.

Referenced by Build().

7.48.3.3 bool clGUILofar::BuildTable2 () [private]

Definition at line 358 of file GUILofar.cc.

References clGtkUtils::BuildOptionMenu(), cpaLTypeMenu, cpaLWindowMenu, cpaLWinLengthMenu, cpLHigherFreqTxt, cpLLowerFreqTxt, cpLOverlapTxt, cpLTypeTxt, cpLWindowTxt, cpLWinLengthTxt, cpLWinParamTxt, _stLofarReq::fHighFreq, _stLofarReq::fLowFreq, GtkUtils, gwaMIType, gwaMIWindow, gwaMIWinLength, gwEHigherFreq, gwELowerFreq, gwEOverlap, gwEWinParam, gwLHigherFreq, gwLLowerFreq, gwLOverlap, gwLType, gwLWindow, gwLWinLength, gwLWinParam, gwMType, gwMWindow, gwMWinLength, gwOMType, gwOMWindow, gwOMWinLength, gwTable2, gwVBox, LGUI_CONV_BUF_SIZE, LGUI_ENTRY_WIDTH, LGUI_PADDING, LGUI_TYPE_ITEMS, LGUI_WIN_LENGTH_ITEMS, LGUI_WINDOW_ITEMS, _stLofarReq::lWinLength, and sLofarRq.

Referenced by Build().

7.48.3.4 bool clGUILofar::BuildTable3 () [private]

Definition at line 505 of file GUILofar.cc.

References clGtkUtils::BuildOptionMenu(), cpaLClipMenu, cpaLPaletteMenu, cpaLRemoveNoiseMenu, cpBSaveTxt, cpCBAverageTxt, cpCBDemonTxt, cpCBLinearTxt, cpCBSavingTxt, cpFSSaveTxt, cpLAlphaTxt, cpLAverageCountTxt, cpLClipTxt, cpLGapLengthTxt, cpLMeanLengthTxt, cpLPaletteTxt, cpLRemoveNoiseTxt, _stLofarReq::fAlpha, GtkUtils, gwaMIClip, gwaMIPalette, gwaMIRemoveNoise, gwBSave, gwCBAverage, gwCBDemon, gwCBLinear, gwCBSaving, gwEAlpha, gwEAverageCount, gwEGapLength, gwEMeanLength, gwFSSave, gwLAlpha, gwLAverageCount, gwLClip, gwLGapLength, gwLMeanLength, gwLPalette, gwLRemoveNoise, gwMClip, gwMPalette, gwMRemoveNoise, gwOMClip, gwOMPalette, gwOMRemoveNoise, gwTable3, gwVBox, _stLofarReq::iClip, iPalette, _stLofarReq::iRemoveNoise, _stLofarReq::lAvgCount, _stLofarReq::lGapLength, LGUI_CLIP_ITEMS, LGUI_CONV_BUF_SIZE, LGUI_ENTRY_WIDTH, LGUI_PADDING, LGUI_PALETTE_ITEMS, LGUI_REMOVE_NOISE_ITEMS, _stLofarReq::lMeanLength, and sLofarRq.

Referenced by Build().

7.48.3.5 bool clGUILofar::BuildTable4 () [private]

Definition at line 705 of file GUILofar.cc.

References cpLClipValueTxt, goAClipValue, gwHSClipValue, gwLClipValue, gwTable4, gwVBox, and LGUI_PADDING.

Referenced by Build().

7.48.3.6 bool clGUILofar::BuildTableLofar () [private]

Definition at line 737 of file GUILofar.cc.

References gwDACursor, gwDALine, gwDALofar, gwHRFrequency, gwLBottomTime, gwLTopTime, gwSWLofar, gwTableLofar, gwTableLofar2, gwVBox, gwVRTime, iFit, LGUI_CURSOR_HEIGHT, LGUI_LINESPECT_HEIGHT, _stLofarReq::lWinLength, and sLofarRq.

Referenced by Build().

7.48.3.7 bool clGUILofar::BuildDrawingPrims () [private]

Definition at line 850 of file GUILofar.cc.

References Cfg, gcCrossHair, clCfgFile::GetInt(), ggcCursorBG, ggcCursorFG, ggcLineBG, ggcLineFG, ggcLofarBG, ggcLofarFG, gwDACursor, gwDALine, gwDALofar, LGUI_CURSOR_BG, LGUI_CURSOR_FG, LGUI_LINE_BG, LGUI_LINE_FG, LGUI_LOFAR_BG, LGUI_LOFAR_FG, and SetPalette().

Referenced by Build().

7.48.3.8 void clGUILofar::FreeDrawingPrims () [private]

Definition at line 902 of file GUILofar.cc.

References gcCrossHair, ggcCursorBG, ggcCursorFG, ggcLineBG, ggcLineFG, ggcLofarBG, and ggcLofarFG.

Referenced by Exec().

7.48.3.9 bool clGUILofar::ConnectSignals () [private]

Definition at line 914 of file GUILofar.cc.

References clGtkUtils::ConnectMotionEvent(), goAClipValue, GtkUtils, gwaMIPalette, gwBConnect, gwBDisconnect, gwBSave, gwCBAverage, gwCBFreeze, gwCBHide, gwCBSaving, gwDACursor, gwDALine, gwDALofar, gwFSSave, gwHRFrequency, gwVRTime, gwWindow, LGUI_PALETTE_ITEMS, WrapOnAverageToggled(), WrapOnClipValueChanged(), WrapOnConfigureLofar(), WrapOnConnectClick(), WrapOnDelete(), WrapOnExposeCursor(), WrapOnExposeLine(), WrapOnExposeLofar(), WrapOnFreezeToggled(), Wrap-

OnHideToggled(), WrapOnMotionLofar(), WrapOnPaletteActivate(), and WrapOnSaveClicks().

Referenced by Exec().

7.48.3.10 bool clGUILofar::ParseServerStr (char *, int *, const char *) [private]

Definition at line 979 of file GUILofar.cc.

References LGUI_SERVER_MAXLEN.

Referenced by OnConnectClick().

7.48.3.11 bool clGUILofar::InitConnection (const char *, int) [private]

Definition at line 1002 of file GUILofar.cc.

References AvgSpectData, bConfigured, bConnected, clSockClie::Connect(), GDT, giGdkTag, GLOBAL_HEADER_LEN, iClips, LGUI_REQ_PROC, l-SpectSize, _stLofarReq::lWinLength, SClient, SendSettings(), clSockOp::Set-Handle(), sLofarRq, SOp, SpectData, WrapOnGdkInput(), and clSock-Op::WriteN().

Referenced by OnConnectClick().

7.48.3.12 bool clGUILofar::SendSettings () [private]

Definition at line 1038 of file GUILofar.cc.

References _stLofarReq::bDemon, _stLofarReq::bLinear, _stLofarReq::fAlpha, _stLofarReq::fHighFreq, _stLofarReq::fLowFreq, _stLofarReq::fWinParameter, GLOBAL_HEADER_LEN, GtkUtils, gwaMIClip, gwaMIRemoveNoise, gwa-MIType, gwaMIWindow, gwaMIWinLength, gwCBDemon, gwCBLinear, gw-EAlpha, gwEAverageCount, gwEGapLength, gwEHigherFreq, gwELower-Freq, gwEMeanLength, gwEOverlap, gwEWinParam, gwOMClip, gw-OMRemoveNoise, gwOMType, gwOMWindow, gwOMWinLength, gw-SBChannel, iBeamCount, _stLofarReq::iChannel, _stLofarReq::iClip, _st-LofarReq::iOverlap, _stLofarReq::iRemoveNoise, _stLofarReq::iType, _stLofar-Req::iWindow, _stLofarReq::lAvgCount, _stLofarReq::lGapLength, LGUI_-CLIP_ITEMS, LGUI_REMOVE_NOISE_ITEMS, LGUI_TYPE_ITEMS, LGUI_-WIN_LENGTH_ITEMS, LGUI_WINDOW_ITEMS, _stLofarReq::lMeanLength, LofarMsg, _stLofarReq::lWinLength, clGtkUtils::OptionMenuGetActive(), cl-LofarMsg::SetRequest(), sLofarRq, SOp, and clSockOp::WriteN().

Referenced by InitConnection().

7.48.3.13 void clGUILofar::PrintStatus () [private]

Definition at line 1097 of file GUILofar.cc.

References _stLofarReq::bDemon, _stLofarRes::fDemonBand, _stLofarRes::fHighFreq, _stLofarRes::fLineTime, _stLofarRes::fLowFreq, _stLofarRes::fPeakLevel, guSbCtxt, gwStatusBar, iClips, iCursorX, iCursorY, _stLofarCursor::iDistance, iFit, iLofarWidth, _stLofarCursor::iPosition, LGUI_CONV_BUF_SIZE, _stLofarRes::lSpectLength, sLCursor, sLofarResHdr, sLofarRq, and _stLofarRes::sTimeStamp.

Referenced by OnGdkInput(), and OnMotionLofar().

7.48.3.14 void clGUILofar::SetPalette (int) [private]

Definition at line 1161 of file GUILofar.cc.

References FBLofar, LGUI_PAL_BW, LGUI_PAL_DIR, LGUI_PAL_GREEN, LGUI_PAL_GREEN2, LGUI_PAL_GREEN3, LGUI_PAL_GREEN4, LGUI_PAL_HSV, LGUI_PAL_LIGHT, LGUI_PAL_PUREGREEN, LGUI_PAL_TEMP, LGUI_PAL_WB, clFrameBuf::PalGenBW(), clFrameBuf::PalGenDir(), clFrameBuf::PalGenGreen(), clFrameBuf::PalGenGreen2(), clFrameBuf::PalGenGreen3(), clFrameBuf::PalGenGreen4(), clFrameBuf::PalGenHSV(), clFrameBuf::PalGenLight(), clFrameBuf::PalGenPureGreen(), clFrameBuf::PalGenTemp(), and clFrameBuf::PalGenWB().

Referenced by BuildDrawingPrims(), and OnPaletteActivate().

7.48.3.15 void clGUILofar::Configure () [private]

Definition at line 1204 of file GUILofar.cc.

References bConfigured, _stLofarReq::bDemon, FBLofar, _stLofarRes::fDemonBand, _stLofarRes::fHighFreq, _stLofarRes::fLineTime, _stLofarRes::fLowFreq, gwDACursor, gwDALine, gwDALofar, gwEHigherFreq, gwELowerFreq, gwHRFrequency, gwSWLofar, gwVRTime, gwWindow, iFit, iLofarHeight, iLofarWidth, LGUI_CONV_BUF_SIZE, _stLofarRes::lSpectLength, lSpectSize, clFrameBuf::SetSize(), sLofarResHdr, and sLofarRq.

Referenced by OnGdkInput().

7.48.3.16 void clGUILofar::ConfigureHeight () [private]

Definition at line 1249 of file GUILofar.cc.

References FBLofar, gwDALofar, and sLofarResHdr.

Referenced by OnGdkInput().

7.48.3.17 void clGUILofar::DrawCursor () [private]

Definition at line 1259 of file GUILofar.cc.

References ggcCursorBG, ggcCursorFG, gwDACursor, _stLofarCursor::iDistance, _stLofarCursor::iPosition, _stLofarCursor::iType, LGUI_CURSOR_11, LGUI_CURSOR_INF, and sLCursor.

Referenced by OnExposeCursor(), and OnMotionLofar().

7.48.3.18 void clGUILofar::SaveInfo (const char *, time_t) [private]

Definition at line 1291 of file GUILofar.cc.

References _stLofarReq::bDemon, FB_TIFF_MODE, _stLofarRes::fDemonBand, _stLofarRes::fHighFreq, _stLofarRes::fLineTime, _stLofarRes::fLowFreq, sLofarResHdr, and sLofarRq.

Referenced by OnSaveClicks(), and StartNewImgFile().

7.48.3.19 void clGUILofar::StartNewImgFile () [private]

Definition at line 1312 of file GUILofar.cc.

References bSaving, FB_TIFF_CONT_VERTICAL, FBLofar, gwCBSaving, iCompressMode, iImgCount, iJPEGQuality, SaveInfo(), clFrameBuf::StartSaveToFile(), clFrameBuf::StopSaveToFile(), and strImgFileName.

Referenced by OnGdkInput().

7.48.3.20 int clGUILofar::Exec ()

Definition at line 1415 of file GUILofar.cc.

References Build(), clFrameBuf::Clear(), ConnectSignals(), FBLofar, FreeDrawingPrims(), iLofarHeight, iLofarWidth, and clFrameBuf::SetSize().

Referenced by main().

7.48.3.21 gint clGUILofar::OnDelete (GtkWidget *, GdkEventAny *)

Definition at line 1435 of file GUILofar.cc.

References bRun, bSaving, FBLofar, and clFrameBuf::StopSaveToFile().

Referenced by WrapOnDelete().

7.48.3.22 void clGUILofar::OnHideToggled (GtkToggleButton *, gpointer)

Definition at line 1447 of file GUILofar.cc.

References gwTable1, gwTable2, gwTable3, and gwTable4.

Referenced by WrapOnHideToggled().

7.48.3.23 gint clGUILofar::OnConnectClick (GtkWidget *, gpointer)

Definition at line 1466 of file GUILofar.cc.

References bConnected, clSockOp::Close(), giGdkTag, gwBConnect, gwCServer, InitConnection(), LGUI_SERVER_MAXLEN, ParseServerStr(), clSockOp::Shutdown(), and SOp.

Referenced by WrapOnConnectClick().

7.48.3.24 void clGUILofar::OnFreezeToggled (GtkToggleButton *, gpointer)

Definition at line 1504 of file GUILofar.cc.

References bFreezed.

Referenced by WrapOnFreezeToggled().

7.48.3.25 gint clGUILofar::OnExposeLofar (GtkWidget *, GdkEventExpose *, gpointer)

Definition at line 1510 of file GUILofar.cc.

References FBLofar, clFrameBuf::GetCurPtr(), clFrameBuf::GetRowStride(), ggcLofarFG, gwDALofar, iLofarHeight, and iLofarWidth.

Referenced by WrapOnExposeLofar().

7.48.3.26 gint clGUILofar::OnConfigureLofar (GtkWidget *, GdkEventConfigure *, gpointer)

Definition at line 1532 of file GUILofar.cc.

Referenced by WrapOnConfigureLofar().

7.48.3.27 gint clGUILofar::OnExposeLine (GtkWidget *, GdkEventExpose *, gpointer)

Definition at line 1540 of file GUILofar.cc.

References GDT, ggcLineFG, gwDALine, lSpectSize, and SpectData.

Referenced by WrapOnExposeLine().

7.48.3.28 gint clGUILofar::OnExposeCursor (GtkWidget *, GdkEventExpose *, gpointer)

Definition at line 1572 of file GUILofar.cc.

References DrawCursor().

Referenced by WrapOnExposeCursor().

7.48.3.29 gint clGUILofar::OnMotionLofar (GtkWidget *, GdkEventMotion *, gpointer)

Definition at line 1580 of file GUILofar.cc.

References bCursorDrag, DrawCursor(), iCursorX, iCursorY, _stLofarCursor::iDistance, _stLofarCursor::iPosition, _stLofarCursor::iType, LGUI_CURSOR_11, LGUI_CURSOR_INF, PrintStatus(), and sLCursor.

Referenced by WrapOnMotionLofar().

7.48.3.30 gint clGUILofar::OnPaletteActivate (GtkWidget *, gpointer)

Definition at line 1626 of file GUILofar.cc.

References GtkUtils, gwaMIPalette, gwOMPalette, iPalette, LGUI_PALETTE_ITEMS, clGtkUtils::OptionMenuGetActive(), and SetPalette().

Referenced by WrapOnPaletteActivate().

7.48.3.31 void clGUILofar::OnAverageToggled (GtkToggleButton *, gpointer)

Definition at line 1635 of file GUILofar.cc.

References bAveraged.

Referenced by WrapOnAverageToggled().

7.48.3.32 void clGUILofar::OnClipValueChanged (GtkAdjustment *, gpointer)

Definition at line 1641 of file GUILofar.cc.

References fClip.

Referenced by WrapOnClipValueChanged().

7.48.3.33 void clGUILofar::OnSaveClicks (GtkWidget *, gpointer)

Definition at line 1647 of file GUILofar.cc.

References _stLofarReq::bDemon, bSaving, FB_TIFF_CONT_VERTICAL, FBLofar, _stLofarRes::fDemonBand, _stLofarRes::fHighFreq, _stLofarRes::fLineTime, _stLofarRes::fLowFreq, clFrameBuf::GetHeight(), gwCBSaving, gwFSSave, iCompressMode, iImgCount, iJPEGQuality, iScanCount, SaveInfo(), clFrameBuf::SaveToFile(), sLofarResHdr, sLofarRq, clFrameBuf::StartSaveToFile(), clFrameBuf::StopSaveToFile(), and strImgFileName.

Referenced by WrapOnSaveClicks().

7.48.3.34 void clGUILofar::OnGdkInput (gpointer, gint, GdkInputCondition)

Definition at line 1746 of file GUILofar.cc.

References AvgSpectData, bAveraged, bConfigured, bConnected, bFreezed, bRun, bSaving, clSockOp::Close(), Configure(), ConfigureHeight(), clFrameBuf::DrawLine(), DSP, FBLofar, fClip, _stLofarRes::fLineTime, _stLofarRes::fPeakLevel, GDT, clFrameBuf::GetCurPtr(), clSockOp::GetErrno(), clLofarMsg::GetResult(), clFrameBuf::GetRowStride(), ggcLineFG, ggcLofarFG, giGdkTag, GLOBAL_HEADER_LEN, gwDALine, gwDALofar, gwLBottomTime, gwLTopTime, iClips, iContSaveScans, iFit, iImgCount, iLofarHeight, iLofarWidth, iResMsgBufSize, iScanCount, LGUI_CONV_BUF_SIZE, LGUI_FIT_AVERAGE, LGUI_FIT_NEIGHBOR, LofarMsg, lSpectSize, PrintStatus(), clSockOp::ReadN(), clSockOp::ReadSelect(), clSockOp::Shutdown(), sLofarResHdr, SOp, SpectData, StartNewImgFile(), and _stLofarRes::sTimeStamp.

Referenced by WrapOnGdkInput().

7.48.4 Member Data Documentation

7.48.4.1 bool clGUILofar::bRun [private]

Definition at line 121 of file GUILofar.hh.

Referenced by clGUILofar(), OnDelete(), and OnGdkInput().

7.48.4.2 bool clGUILofar::bConnected [private]

Definition at line 122 of file GUILofar.hh.

Referenced by clGUILofar(), InitConnection(), OnConnectClick(), and OnGdkInput().

7.48.4.3 bool clGUILofar::bFreezed [private]

Definition at line 123 of file GUILofar.hh.

Referenced by clGUILofar(), OnFreezeToggled(), and OnGdkInput().

7.48.4.4 bool clGUILofar::bConfigured [private]

Definition at line 124 of file GUILofar.hh.

Referenced by clGUILofar(), Configure(), InitConnection(), and OnGdkInput().

7.48.4.5 bool clGUILofar::bAveraged [private]

Definition at line 125 of file GUILofar.hh.

Referenced by clGUILofar(), OnAverageToggled(), and OnGdkInput().

7.48.4.6 bool clGUILofar::bCursorDrag [private]

Definition at line 126 of file GUILofar.hh.

Referenced by clGUILofar(), and OnMotionLofar().

7.48.4.7 bool clGUILofar::bSaving [private]

Definition at line 127 of file GUILofar.hh.

Referenced by OnDelete(), OnGdkInput(), OnSaveClicks(), and StartNewImgFile().

7.48.4.8 int clGUILofar::iResMsgBufSize [private]

Definition at line 128 of file GUILofar.hh.

Referenced by OnGdkInput().

7.48.4.9 int clGUILofar::iFit [private]

Definition at line 129 of file GUILofar.hh.

Referenced by BuildTableLofar(), clGUILofar(), Configure(), OnGdkInput(), and PrintStatus().

7.48.4.10 int clGUILofar::iPalette [private]

Definition at line 130 of file GUILofar.hh.

Referenced by BuildTable3(), clGUILofar(), and OnPaletteActivate().

7.48.4.11 int clGUILofar::iLofarWidth [private]

Definition at line 131 of file GUILofar.hh.

Referenced by clGUILofar(), Configure(), Exec(), OnExposeLofar(), OnGdkInput(), and PrintStatus().

7.48.4.12 int clGUILofar::iLofarHeight [private]

Definition at line 132 of file GUILofar.hh.

Referenced by clGUILofar(), Configure(), Exec(), OnExposeLofar(), and OnGdkInput().

7.48.4.13 int clGUILofar::iCursorX [private]

Definition at line 133 of file GUILofar.hh.

Referenced by OnMotionLofar(), and PrintStatus().

7.48.4.14 int clGUILofar::iCursorY [private]

Definition at line 134 of file GUILofar.hh.

Referenced by OnMotionLofar(), and PrintStatus().

7.48.4.15 int clGUILofar::iClips [private]

Definition at line 135 of file GUILofar.hh.

Referenced by InitConnection(), OnGdkInput(), and PrintStatus().

7.48.4.16 int clGUILofar::iTIFFCompression [private]

Definition at line 136 of file GUILofar.hh.

Referenced by clGUILofar().

7.48.4.17 int clGUILofar::iJPEGQuality [private]

Definition at line 137 of file GUILofar.hh.

Referenced by clGUILofar(), OnSaveClicks(), and StartNewImgFile().

7.48.4.18 int clGUILofar::iContSaveScans [private]

Definition at line 138 of file GUILofar.hh.

Referenced by clGUILofar(), and OnGdkInput().

7.48.4.19 int clGUILofar::iCompressMode [private]

Definition at line 139 of file GUILofar.hh.

Referenced by clGUILofar(), OnSaveClicks(), and StartNewImgFile().

7.48.4.20 int clGUILofar::iScanCount [private]

Definition at line 140 of file GUILofar.hh.

Referenced by OnGdkInput(), and OnSaveClicks().

7.48.4.21 int clGUILofar::iImgCount [private]

Definition at line 141 of file GUILofar.hh.

Referenced by OnGdkInput(), OnSaveClicks(), and StartNewImgFile().

7.48.4.22 int clGUILofar::iBeamCount [private]

Definition at line 142 of file GUILofar.hh.

Referenced by BuildTable1(), clGUILofar(), and SendSettings().

7.48.4.23 long clGUILofar::lSpectSize [private]

Definition at line 143 of file GUILofar.hh.

Referenced by Configure(), InitConnection(), OnExposeLine(), and OnGdk-Input().

7.48.4.24 GDT clGUILofar::fClip [private]

Definition at line 144 of file GUILofar.hh.

Referenced by clGUILofar(), OnClipValueChanged(), and OnGdkInput().

7.48.4.25 stLofarReq clGUILofar::sLofarRq [private]

Definition at line 145 of file GUILofar.hh.

Referenced by BuildTable2(), BuildTable3(), BuildTableLofar(), clGUILofar(), Configure(), InitConnection(), OnSaveClicks(), PrintStatus(), SaveInfo(), and SendSettings().

7.48.4.26 stLofarRes clGUILofar::sLofarResHdr [private]

Definition at line 146 of file GUILofar.hh.

Referenced by Configure(), ConfigureHeight(), OnGdkInput(), OnSaveClicks(), PrintStatus(), and SaveInfo().

7.48.4.27 stLofarCursor clGUILofar::sLCursor [private]

Definition at line 147 of file GUILofar.hh.

Referenced by clGUILofar(), DrawCursor(), OnMotionLofar(), and Print-Status().

7.48.4.28 GList* clGUILofar::glServer [private]

Definition at line 149 of file GUILofar.hh.

Referenced by BuildTable1(), and clGUILofar().

7.48.4.29 gint clGUILofar::giGdkTag [private]

Definition at line 151 of file GUILofar.hh.

Referenced by InitConnection(), OnConnectClick(), and OnGdkInput().

7.48.4.30 guint clGUILofar::guSbCtxt [private]

Definition at line 152 of file GUILofar.hh.

Referenced by Build(), and PrintStatus().

7.48.4.31 GtkWidget* clGUILofar::gwWindow [private]

Definition at line 154 of file GUILofar.hh.

Referenced by Build(), Configure(), and ConnectSignals().

7.48.4.32 GtkWidget* clGUILofar::gwVBox [private]

Definition at line 155 of file GUILofar.hh.

Referenced by Build(), BuildTable1(), BuildTable2(), BuildTable3(), BuildTable4(), and BuildTableLofar().

7.48.4.33 GtkWidget* clGUILofar::gwCBHide [private]

Definition at line 156 of file GUILofar.hh.

Referenced by Build(), and ConnectSignals().

7.48.4.34 GtkWidget* clGUILofar::gwStatusBar [private]

Definition at line 157 of file GUILofar.hh.

Referenced by Build(), and PrintStatus().

7.48.4.35 GtkWidget* clGUILofar::gwTable1 [private]

Definition at line 159 of file GUILofar.hh.

Referenced by BuildTable1(), and OnHideToggled().

7.48.4.36 GtkWidget* clGUILofar::gwLServer [private]

Definition at line 160 of file GUILofar.hh.

Referenced by BuildTable1().

7.48.4.37 GtkWidget* clGUILofar::gwCServer [private]

Definition at line 161 of file GUILofar.hh.

Referenced by BuildTable1(), and OnConnectClick().

7.48.4.38 GtkWidget* clGUILofar::gwLChannel [private]

Definition at line 162 of file GUILofar.hh.

Referenced by BuildTable1().

7.48.4.39 GtkObject* clGUILofar::goAChannel [private]

Definition at line 163 of file GUILofar.hh.

Referenced by BuildTable1().

7.48.4.40 GtkWidget* clGUILofar::gwSBChannel [private]

Definition at line 164 of file GUILofar.hh.

Referenced by BuildTable1(), and SendSettings().

7.48.4.41 GtkWidget* clGUILofar::gwBConnect [private]

Definition at line 165 of file GUILofar.hh.

Referenced by BuildTable1(), ConnectSignals(), and OnConnectClick().

7.48.4.42 GtkWidget* clGUILofar::gwBDisconnect [private]

Definition at line 166 of file GUILofar.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.48.4.43 GtkWidget* clGUILofar::gwCBFreeze [private]

Definition at line 167 of file GUILofar.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.48.4.44 GtkWidget* clGUILofar::gwTable2 [private]

Definition at line 169 of file GUILofar.hh.

Referenced by BuildTable2(), and OnHideToggled().

7.48.4.45 GtkWidget* clGUILofar::gwLType [private]

Definition at line 170 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.46 GtkWidget* clGUILofar::gwOMType [private]

Definition at line 171 of file GUILofar.hh.

Referenced by BuildTable2(), and SendSettings().

7.48.4.47 GtkWidget* clGUILofar::gwMType [private]

Definition at line 172 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.48 GtkWidget* clGUILofar::gwaMIType[LGUI_TYPE_ITEMS] [private]

Definition at line 173 of file GUILofar.hh.

Referenced by BuildTable2(), and SendSettings().

7.48.4.49 GtkWidget* clGUILofar::gwLWindow [private]

Definition at line 174 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.50 GtkWidget* clGUILofar::gwOMWindow [private]

Definition at line 175 of file GUILofar.hh.

Referenced by BuildTable2(), and SendSettings().

7.48.4.51 GtkWidget* clGUILofar::gwMWindow [private]

Definition at line 176 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.52 GtkWidget* clGUILofar::gwaMIWindow[LGUI_WINDOW_ITEMS] [private]

Definition at line 177 of file GUILofar.hh.

Referenced by BuildTable2(), and SendSettings().

7.48.4.53 GtkWidget* clGUILofar::gwLWinParam [private]

Definition at line 178 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.54 GtkWidget* clGUILofar::gwEWinParam [private]

Definition at line 179 of file GUILofar.hh.

Referenced by BuildTable2(), and SendSettings().

7.48.4.55 GtkWidget* clGUILofar::gwLWinLength [private]

Definition at line 180 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.56 GtkWidget* clGUILofar::gwOMWinLength [private]

Definition at line 181 of file GUILofar.hh.

Referenced by BuildTable2(), and SendSettings().

7.48.4.57 GtkWidget* clGUILofar::gwMWinLength [private]

Definition at line 182 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.58 GtkWidget* clGUILofar::gwaMIWinLength[LGUI_WIN_LENGTH_ITEMS] [private]

Definition at line 183 of file GUILofar.hh.

Referenced by BuildTable2(), and SendSettings().

7.48.4.59 GtkWidget* clGUILofar::gwLLowerFreq [private]

Definition at line 184 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.60 GtkWidget* clGUILofar::gwELowerFreq [private]

Definition at line 185 of file GUILofar.hh.

Referenced by BuildTable2(), Configure(), and SendSettings().

7.48.4.61 GtkWidget* clGUILofar::gwLHigherFreq [private]

Definition at line 186 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.62 GtkWidget* clGUILofar::gwEHigherFreq [private]

Definition at line 187 of file GUILofar.hh.

Referenced by BuildTable2(), Configure(), and SendSettings().

7.48.4.63 GtkWidget* clGUILofar::gwLOverlap [private]

Definition at line 188 of file GUILofar.hh.

Referenced by BuildTable2().

7.48.4.64 GtkWidget* clGUILofar::gwEOverlap [private]

Definition at line 189 of file GUILofar.hh.

Referenced by BuildTable2(), and SendSettings().

7.48.4.65 GtkWidget* clGUILofar::gwTable3 [private]

Definition at line 191 of file GUILofar.hh.

Referenced by BuildTable3(), and OnHideToggled().

7.48.4.66 GtkWidget* clGUILofar::gwLRemoveNoise [private]

Definition at line 192 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.67 GtkWidget* clGUILofar::gwOMRemoveNoise [private]

Definition at line 193 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.68 GtkWidget* clGUILofar::gwMRemoveNoise [private]

Definition at line 194 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.69 GtkWidget* clGUILofar::gwaMIRemoveNoise[LGUI_REMOVE_NOISE_ITEMS] [private]

Definition at line 195 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.70 GtkWidget* clGUILofar::gwLAlpha [private]

Definition at line 196 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.71 GtkWidget* clGUILofar::gwEAlpha [private]

Definition at line 197 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.72 GtkWidget* clGUILofar::gwLMeanLength [private]

Definition at line 198 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.73 GtkWidget* clGUILofar::gwEMeanLength [private]

Definition at line 199 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.74 GtkWidget* clGUILofar::gwLGapLength [private]

Definition at line 200 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.75 GtkWidget* clGUILofar::gwEGapLength [private]

Definition at line 201 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.76 GtkWidget* clGUILofar::gwLAverageCount [private]

Definition at line 202 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.77 GtkWidget* clGUILofar::gwEAverageCount [private]

Definition at line 203 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.78 GtkWidget* clGUILofar::gwLClip [private]

Definition at line 204 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.79 GtkWidget* clGUILofar::gwOMClip [private]

Definition at line 205 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.80 GtkWidget* clGUILofar::gwMClip [private]

Definition at line 206 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.81 GtkWidget* clGUILofar::gwaMIClip[LGUI_CLIP_ITEMS] [private]

Definition at line 207 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.82 GtkWidget* clGUILofar::gwCBLinear [private]

Definition at line 208 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.83 GtkWidget* clGUILofar::gwCBDemon [private]

Definition at line 209 of file GUILofar.hh.

Referenced by BuildTable3(), and SendSettings().

7.48.4.84 GtkWidget* clGUILofar::gwLPalette [private]

Definition at line 210 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.85 GtkWidget* clGUILofar::gwOMPalette [private]

Definition at line 211 of file GUILofar.hh.

Referenced by BuildTable3(), and OnPaletteActivate().

7.48.4.86 GtkWidget* clGUILofar::gwMPalette [private]

Definition at line 212 of file GUILofar.hh.

Referenced by BuildTable3().

7.48.4.87 GtkWidget* clGUILofar::gwaMIPalette[LGUI_PALETTE_ITEMS] [private]

Definition at line 213 of file GUILofar.hh.

Referenced by BuildTable3(), ConnectSignals(), and OnPaletteActivate().

7.48.4.88 GtkWidget* clGUILofar::gwCBAverage [private]

Definition at line 214 of file GUILofar.hh.

Referenced by BuildTable3(), and ConnectSignals().

7.48.4.89 GtkWidget* clGUILofar::gwCBSaving [private]

Definition at line 215 of file GUILofar.hh.

Referenced by BuildTable3(), ConnectSignals(), OnSaveClicks(), and StartNewImgFile().

7.48.4.90 GtkWidget* clGUILofar::gwBSave [private]

Definition at line 216 of file GUILofar.hh.

Referenced by BuildTable3(), and ConnectSignals().

7.48.4.91 GtkWidget* clGUILofar::gwFSSave [private]

Definition at line 217 of file GUILofar.hh.

Referenced by BuildTable3(), ConnectSignals(), and OnSaveClicks().

7.48.4.92 GtkWidget* clGUILofar::gwTable4 [private]

Definition at line 219 of file GUILofar.hh.

Referenced by BuildTable4(), and OnHideToggled().

7.48.4.93 GtkWidget* clGUILofar::gwLClipValue [private]

Definition at line 220 of file GUILofar.hh.

Referenced by BuildTable4().

7.48.4.94 GtkObject* clGUILofar::goAClipValue [private]

Definition at line 221 of file GUILofar.hh.

Referenced by BuildTable4(), and ConnectSignals().

7.48.4.95 GtkWidget* clGUILofar::gwHSClipValue [private]

Definition at line 222 of file GUILofar.hh.

Referenced by BuildTable4().

7.48.4.96 GtkWidget* clGUILofar::gwTableLofar [private]

Definition at line 224 of file GUILofar.hh.

Referenced by BuildTableLofar().

7.48.4.97 GtkWidget* clGUILofar::gwLTopTime [private]

Definition at line 225 of file GUILofar.hh.

Referenced by BuildTableLofar(), and OnGdkInput().

7.48.4.98 GtkWidget* clGUILofar::gwLBottomTime [private]

Definition at line 226 of file GUILofar.hh.

Referenced by BuildTableLofar(), and OnGdkInput().

7.48.4.99 GtkWidget* clGUILofar::gwSWLofar [private]

Definition at line 227 of file GUILofar.hh.

Referenced by BuildTableLofar(), and Configure().

7.48.4.100 GtkWidget* clGUILofar::gwTableLofar2 [private]

Definition at line 228 of file GUILofar.hh.

Referenced by BuildTableLofar().

7.48.4.101 GtkWidget* clGUILofar::gwHRFrequency [private]

Definition at line 229 of file GUILofar.hh.

Referenced by BuildTableLofar(), Configure(), and ConnectSignals().

7.48.4.102 GtkWidget* clGUILofar::gwVRTime [private]

Definition at line 230 of file GUILofar.hh.

Referenced by BuildTableLofar(), Configure(), and ConnectSignals().

7.48.4.103 GtkWidget* clGUILofar::gwDALine [private]

Definition at line 231 of file GUILofar.hh.

Referenced by Build(), BuildDrawingPrims(), BuildTableLofar(), Configure(), ConnectSignals(), OnExposeLine(), and OnGdkInput().

7.48.4.104 GtkWidget* clGUILofar::gwDACursor [private]

Definition at line 232 of file GUILofar.hh.

Referenced by BuildDrawingPrims(), BuildTableLofar(), Configure(), ConnectSignals(), and DrawCursor().

7.48.4.105 GtkWidget* clGUILofar::gwDALofar [private]

Definition at line 233 of file GUILofar.hh.

Referenced by Build(), BuildDrawingPrims(), BuildTableLofar(), Configure(), ConfigureHeight(), ConnectSignals(), OnExposeLofar(), and OnGdkInput().

7.48.4.106 GdkGC* clGUILofar::ggcLofarBG [private]

Definition at line 235 of file GUILofar.hh.

Referenced by BuildDrawingPrims(), and FreeDrawingPrims().

7.48.4.107 GdkGC* clGUILofar::ggcLofarFG [private]

Definition at line 236 of file GUILofar.hh.

Referenced by BuildDrawingPrims(), FreeDrawingPrims(), OnExposeLofar(), and OnGdkInput().

7.48.4.108 GdkGC∗ clGUILofar::ggcLineBG [private]

Definition at line 237 of file GUILofar.hh.

Referenced by BuildDrawingPrims(), and FreeDrawingPrims().

7.48.4.109 GdkGC∗ clGUILofar::ggcLineFG [private]

Definition at line 238 of file GUILofar.hh.

Referenced by BuildDrawingPrims(), FreeDrawingPrims(), OnExposeLine(), and OnGdkInput().

7.48.4.110 GdkGC∗ clGUILofar::ggcCursorBG [private]

Definition at line 239 of file GUILofar.hh.

Referenced by BuildDrawingPrims(), DrawCursor(), and FreeDrawingPrims().

7.48.4.111 GdkGC∗ clGUILofar::ggcCursorFG [private]

Definition at line 240 of file GUILofar.hh.

Referenced by BuildDrawingPrims(), DrawCursor(), and FreeDrawingPrims().

7.48.4.112 GdkCursor∗ clGUILofar::gcCrossHair [private]

Definition at line 241 of file GUILofar.hh.

Referenced by Build(), BuildDrawingPrims(), and FreeDrawingPrims().

7.48.4.113 std::string clGUILofar::strImgFileName [private]

Definition at line 243 of file GUILofar.hh.

Referenced by OnSaveClicks(), and StartNewImgFile().

7.48.4.114 clAlloc clGUILofar::SpectData [private]

Definition at line 244 of file GUILofar.hh.

Referenced by InitConnection(), OnExposeLine(), and OnGdkInput().

7.48.4.115 clAlloc clGUILofar::AvgSpectData [private]

Definition at line 245 of file GUILofar.hh.

Referenced by InitConnection(), and OnGdkInput().

7.48.4.116 clCfgFile clGUILofar::Cfg [private]

Definition at line 246 of file GUILofar.hh.

Referenced by BuildDrawingPrims(), and clGUILofar().

7.48.4.117 clDSPOp clGUILofar::DSP [private]

Definition at line 247 of file GUILofar.hh.

Referenced by OnGdkInput().

7.48.4.118 clFrameBuf clGUILofar::FBLofar [private]

Definition at line 248 of file GUILofar.hh.

Referenced by Configure(), ConfigureHeight(), Exec(), OnDelete(), OnExposeLofar(), OnGdkInput(), OnSaveClicks(), SetPalette(), and StartNewImgFile().

7.48.4.119 clGtkUtils clGUILofar::GtkUtils [private]

Definition at line 249 of file GUILofar.hh.

Referenced by BuildTable1(), BuildTable2(), BuildTable3(), ConnectSignals(), OnPaletteActivate(), and SendSettings().

7.48.4.120 clLofarMsg clGUILofar::LofarMsg [private]

Definition at line 250 of file GUILofar.hh.

Referenced by OnGdkInput(), and SendSettings().

7.48.4.121 clSockClie clGUILofar::SClient [private]

Definition at line 251 of file GUILofar.hh.

Referenced by InitConnection().

7.48.4.122 clSockOp clGUILofar::SOp [private]

Definition at line 252 of file GUILofar.hh.

Referenced by InitConnection(), OnConnectClick(), OnGdkInput(), and SendSettings().

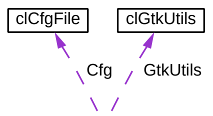
The documentation for this class was generated from the following files:

- **GUILofar.hh**
- **GUILofar.cc**

7.49 clGUITransient Class Reference

`#include <GUITrans.hh>`

Collaboration diagram for clGUITransient:



Public Member Functions

- **clGUITransient** ()
- **~clGUITransient** ()
- int **Main** (int ∗, char ∗∗∗)
- gboolean **OnDelete** (GtkWidget ∗, GdkEvent ∗, gpointer)
- void **OnConnectClick** (GtkButton ∗, gpointer)

Private Member Functions

- bool **GetCfg** ()
- bool **Build** ()
- bool **BuildTable1** ()
- bool **BuildDrawingPrims** ()
- bool **ConnectSignals** ()

Private Attributes

- bool **bRun**
- bool **bConnected**
- gint **iGdkInputTag**
- GList ∗ **glServer**
- GtkWidget ∗ **gwWindow**
- GtkWidget ∗ **gwVBox**
- GtkWidget ∗ **gwTable1**
- GtkWidget ∗ **gwLServer**
- GtkWidget ∗ **gwCServer**
- GtkWidget ∗ **gwBConnect**
- **clCfgFile Cfg**
- **clGtkUtils GtkUtils**

7.49.1 Constructor & Destructor Documentation

7.49.1.1 clGUITransient::clGUITransient ()

Definition at line 154 of file GUITrans.cc.

References bConnected, and bRun.

7.49.1.2 clGUITransient::~clGUITransient ()

Definition at line 161 of file GUITrans.cc.

7.49.2 Member Function Documentation

7.49.2.1 bool clGUITransient::GetCfg () `[private]`

Definition at line 66 of file GUITrans.cc.

References Cfg, GUITRANS_CFGFILE, and clCfgFile::SetFileName().

Referenced by Main().

7.49.2.2 bool clGUITransient::Build () `[private]`

Definition at line 73 of file GUITrans.cc.

References BuildDrawingPrims(), BuildTable1(), ConnectSignals(), cpWindowTxt, GUITRANS_WSPACING, gwVBox, and gwWindow.

Referenced by Main().

7.49.2.3 bool clGUITransient::BuildTable1 () `[private]`

Definition at line 92 of file GUITrans.cc.

References clGtkUtils::ComboListFromFile(), cpBConnectTxt, cpLServerTxt, glServer, GtkUtils, GUITRANS_HOSTFILE, GUITRANS_SERVER_MAXLEN, GUITRANS_WSPACING, gwBConnect, gwCServer, gwLServer, gwTable1, and gwVBox.

Referenced by Build().

7.49.2.4 bool clGUITransient::BuildDrawingPrims () `[private]`

Definition at line 148 of file GUITrans.cc.

Referenced by Build().

7.49.2.5 bool clGUITransient::ConnectSignals () [private]

Definition at line 136 of file GUITrans.cc.

References gwBConnect, gwWindow, WrapOnConnectClick(), and WrapOnDelete().

Referenced by Build().

7.49.2.6 int clGUITransient::Main (int ∗, char ∗∗∗)

Definition at line 166 of file GUITrans.cc.

References Build(), cpWindowTxt, GetCfg(), GUITRANS_VER_MAJ, GUITRANS_VER_MIN, and GUITRANS_VER_PL.

Referenced by main().

7.49.2.7 gboolean clGUITransient::OnDelete (GtkWidget ∗, GdkEvent ∗, gpointer)

Definition at line 185 of file GUITrans.cc.

References bRun.

Referenced by WrapOnDelete().

7.49.2.8 void clGUITransient::OnConnectClick (GtkButton ∗, gpointer)

Definition at line 194 of file GUITrans.cc.

Referenced by WrapOnConnectClick().

7.49.3 Member Data Documentation

7.49.3.1 bool clGUITransient::bRun [private]

Definition at line 64 of file GUITrans.hh.

Referenced by clGUITransient(), and OnDelete().

7.49.3.2 bool clGUITransient::bConnected [private]

Definition at line 65 of file GUITrans.hh.

Referenced by clGUITransient().

7.49.3.3 gint clGUITransient::iGdkInputTag [private]

Definition at line 67 of file GUITrans.hh.

7.49.3.4 GList* clGUITransient::glServer [private]

Definition at line 68 of file GUITrans.hh.

Referenced by BuildTable1().

7.49.3.5 GtkWidget* clGUITransient::gwWindow [private]

Definition at line 74 of file GUITrans.hh.

Referenced by Build(), and ConnectSignals().

7.49.3.6 GtkWidget* clGUITransient::gwVBox [private]

Definition at line 75 of file GUITrans.hh.

Referenced by Build(), and BuildTable1().

7.49.3.7 GtkWidget* clGUITransient::gwTable1 [private]

Definition at line 77 of file GUITrans.hh.

Referenced by BuildTable1().

7.49.3.8 GtkWidget* clGUITransient::gwLServer [private]

Definition at line 78 of file GUITrans.hh.

Referenced by BuildTable1().

7.49.3.9 GtkWidget* clGUITransient::gwCServer [private]

Definition at line 79 of file GUITrans.hh.

Referenced by BuildTable1().

7.49.3.10 GtkWidget* clGUITransient::gwBConnect [private]

Definition at line 80 of file GUITrans.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.49.3.11 clCfgFile clGUITransient::Cfg [private]

Definition at line 82 of file GUITrans.hh.

Referenced by GetCfg().

7.49.3.12 clGtkUtils clGUITransient::GtkUtils [private]

Definition at line 83 of file GUITrans.hh.

Referenced by BuildTable1().

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

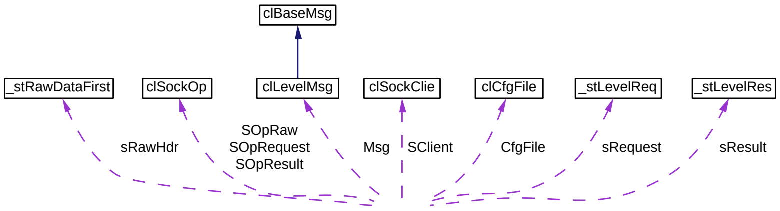
- **GUITrans.hh**
- **GUITrans.cc**

7.50 clLevel Class Reference

Level server.

`#include <Level.hh>`

Collaboration diagram for clLevel:



Public Member Functions

- **clLevel** (int, int)
- ∼**clLevel** ()
- int **Main** (int, char ∗∗)

Private Member Functions

- bool **GetCfg** ()
- bool **GetRequest** ()
- bool **ConnectStream** ()
- bool **Initialize** ()
- bool **ReceiveData** ()
- bool **Process** ()
- bool **SendResult** ()

Private Attributes

- bool **bRun**
- long **lFilterSize**
- long **lRawDataCount**
- long **lRawBufSize**
- long **lSampleCount**
- char **cpStreamSocket** [_POSIX_PATH_MAX+1]
- GDT **fResPeakLevel**
- **stRawDataFirst sRawHdr**
- **stLevelReq sRequest**
- **stLevelRes sResult**
- clDSPAlloc **RawBuf**

- clDSPAlloc **DataBuf**
- **clCfgFile CfgFile**
- clDSPOp **DSP**
- clFilter **Filter**
- **clSockClie SClient**
- **clSockOp SOpRequest**
- **clSockOp SOpResult**
- **clSockOp SOpRaw**
- **clLevelMsg Msg**

7.50.1 Detailed Description

Level server.

Definition at line 46 of file Level.hh.

7.50.2 Constructor & Destructor Documentation

7.50.2.1 clLevel::clLevel (int, int)

Definition at line 254 of file Level.cc.

References bRun, clSockOp::SetHandle(), SOpRequest, and SOpResult.

7.50.2.2 clLevel::∼clLevel ()

Definition at line 262 of file Level.cc.

7.50.3 Member Function Documentation

7.50.3.1 bool clLevel::GetCfg () `[private]`

Definition at line 82 of file Level.cc.

References CfgFile, cpStreamSocket, clCfgFile::GetInt(), clCfgFile::GetStr(), LEVEL_CFGFILE, LEVEL_DEF_FILTER_SIZE, lFilterSize, SD_CFGFILE, and clCfgFile::SetFileName().

7.50.3.2 bool clLevel::GetRequest () `[private]`

Definition at line 93 of file Level.cc.

References clLevelMsg::GetRequest(), GLOBAL_HEADER_LEN, LEVEL_1STREQ_TIMEOUT, Msg, clSockOp::ReadN(), clSockOp::ReadSelect(), SOpRequest, and sRequest.

7.50.3.3 bool clLevel::ConnectStream () [private]

Definition at line 105 of file Level.cc.

References clSockClie::Connect(), cpStreamSocket, _stLevelReq::iChannel, _stRawDataReq::iChannel, LEVEL_RAW1ST_TIMEOUT, clSockOp::ReadN(), clSockOp::ReadSelect(), SClient, clSockOp::SetHandle(), SOpRaw, sRawHdr, sRequest, and clSockOp::WriteN().

7.50.3.4 bool clLevel::Initialize () [private]

Definition at line 123 of file Level.cc.

References bDebug, DataBuf, _stRawDataFirst::dSampleRate, _stLevelReq::fHighFrequency, Filter, _stLevelRes::fIntegrationTime, _stLevelReq::fIntegrationTime, _stLevelReq::fLowFrequency, GDT, _stLevelReq::iAlgorithm, lFilterSize, lRawBufSize, lRawDataCount, lSampleCount, MSG_LEVEL_ALG_MEAN, MSG_LEVEL_ALG_MEDIAN, MSG_LEVEL_ALG_PEAK, MSG_LEVEL_ALG_RMS, MSG_LEVEL_ALG_STDDEV, RawBuf, sRawHdr, sRequest, and sResult.

7.50.3.5 bool clLevel::ReceiveData () [private]

Definition at line 187 of file Level.cc.

References DSP, Filter, fResPeakLevel, GDT, LEVEL_TIMEOUT, lRawBufSize, lRawDataCount, RawBuf, clSockOp::ReadN(), clSockOp::ReadSelect(), and SOpRaw.

7.50.3.6 bool clLevel::Process () [private]

Definition at line 206 of file Level.cc.

References DataBuf, DSP, _stLevelRes::fResult, GDT, _stLevelReq::iAlgorithm, lSampleCount, MSG_LEVEL_ALG_MEAN, MSG_LEVEL_ALG_MEDIAN, MSG_LEVEL_ALG_PEAK, MSG_LEVEL_ALG_RMS, MSG_LEVEL_ALG_STDDEV, sRequest, sResult, and _stLevelRes::sTimeStamp.

7.50.3.7 bool clLevel::SendResult () [private]

Definition at line 242 of file Level.cc.

References _stLevelRes::fPeakLevel, GLOBAL_HEADER_LEN, Msg, clLevelMsg::SetResult(), SOpResult, sResult, and clSockOp::WriteN().

7.50.3.8 int clLevel::Main (int, char **)

Referenced by main().

7.50.4 Member Data Documentation

7.50.4.1 bool clLevel::bRun [private]

Definition at line 48 of file Level.hh.

Referenced by clLevel().

7.50.4.2 long clLevel::lFilterSize [private]

Definition at line 49 of file Level.hh.

Referenced by GetCfg(), and Initialize().

7.50.4.3 long clLevel::lRawDataCount [private]

Definition at line 50 of file Level.hh.

Referenced by Initialize(), and ReceiveData().

7.50.4.4 long clLevel::lRawBufSize [private]

Definition at line 51 of file Level.hh.

Referenced by Initialize(), and ReceiveData().

7.50.4.5 long clLevel::lSampleCount [private]

Definition at line 52 of file Level.hh.

Referenced by Initialize(), and Process().

7.50.4.6 char clLevel::cpStreamSocket[_POSIX_PATH_MAX + 1] [private]

Definition at line 53 of file Level.hh.

Referenced by ConnectStream(), and GetCfg().

7.50.4.7 GDT clLevel::fResPeakLevel [private]

Definition at line 54 of file Level.hh.

Referenced by ReceiveData().

7.50.4.8 stRawDataFirst clLevel::sRawHdr [private]

Definition at line 55 of file Level.hh.

Referenced by ConnectStream(), and Initialize().

7.50.4.9 stLevelReq clLevel::sRequest [private]

Definition at line 56 of file Level.hh.

Referenced by ConnectStream(), GetRequest(), Initialize(), and Process().

7.50.4.10 stLevelRes clLevel::sResult [private]

Definition at line 57 of file Level.hh.

Referenced by Initialize(), Process(), and SendResult().

7.50.4.11 clDSPAlloc clLevel::RawBuf [private]

Definition at line 58 of file Level.hh.

Referenced by Initialize(), and ReceiveData().

7.50.4.12 clDSPAlloc clLevel::DataBuf [private]

Definition at line 59 of file Level.hh.

Referenced by Initialize(), and Process().

7.50.4.13 clCfgFile clLevel::CfgFile [private]

Definition at line 60 of file Level.hh.

Referenced by GetCfg().

7.50.4.14 clDSPOp clLevel::DSP [private]

Definition at line 61 of file Level.hh.

Referenced by Process(), and ReceiveData().

7.50.4.15 clFilter clLevel::Filter [private]

Definition at line 62 of file Level.hh.

Referenced by Initialize(), and ReceiveData().

7.50.4.16 clSockClie clLevel::SClient [private]

Definition at line 63 of file Level.hh.

Referenced by ConnectStream().

7.50.4.17 clSockOp clLevel::SOpRequest [private]

Definition at line 64 of file Level.hh.

Referenced by clLevel(), and GetRequest().

7.50.4.18 clSockOp clLevel::SOpResult [private]

Definition at line 65 of file Level.hh.

Referenced by clLevel(), and SendResult().

7.50.4.19 clSockOp clLevel::SOpRaw [private]

Definition at line 66 of file Level.hh.

Referenced by ConnectStream(), and ReceiveData().

7.50.4.20 clLevelMsg clLevel::Msg [private]

Definition at line 67 of file Level.hh.

Referenced by GetRequest(), and SendResult().

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

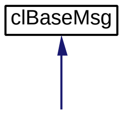
- **Level.hh**
- **Level.cc**

7.51 clLevelMsg Class Reference

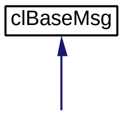
Level server communication.

#include <Messages.hh>

Inheritance diagram for clLevelMsg:



Collaboration diagram for clLevelMsg:



Public Member Functions

- void **SetRequest** (char ∗, const **stpLevelReq**)
- void **GetRequest** (const char ∗, **stpLevelReq**)
- void **SetResult** (void ∗, const **stpLevelRes**)
- void **GetResult** (const void ∗, **stpLevelRes**)

7.51.1 Detailed Description

Level server communication.

Definition at line 588 of file Messages.hh.

7.51.2 Member Function Documentation

7.51.2.1 void clLevelMsg::SetRequest (char ∗, const *stpLevelReq*)

Definition at line 1219 of file Messages.cc.

Referenced by clGUILevel::SendSettings().

7.51.2.2 void clLevelMsg::GetRequest (const char ∗, stpLevelReq)

Definition at line 1239 of file Messages.cc.

References _stLevelReq::fHighFrequency, _stLevelReq::fIntegrationTime, _stLevelReq::fLowFrequency, _stLevelReq::iAlgorithm, and _stLevelReq::iChannel.

Referenced by clLevel::GetRequest().

7.51.2.3 void clLevelMsg::SetResult (void *, const *stpLevelRes*)

Definition at line 1250 of file Messages.cc.

References _stLevelRes::fIntegrationTime, _stLevelRes::fPeakLevel, _stLevelRes::fResult, GLOBAL_HEADER_LEN, and _stLevelRes::sTimeStamp.

Referenced by clLevel::SendResult().

7.51.2.4 void clLevelMsg::GetResult (const void *, stpLevelRes)

Definition at line 1272 of file Messages.cc.

References _stLevelRes::fIntegrationTime, _stLevelRes::fPeakLevel, _stLevelRes::fResult, and _stLevelRes::sTimeStamp.

Referenced by clGUILevel::OnGdkInput().

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

- **Messages.hh**
- **Messages.cc**

7.52 _stLevelReq Struct Reference

Level: request.

`#include <Messages.hh>`

Public Attributes

- int **iChannel**

  Channel.

- int **iAlgorithm**

  Algorithm.

- float **fIntegrationTime**

  Itegration time (s).

- float **fLowFrequency**

  Lower frequency limit (Hz).

- float **fHighFrequency**

  Higer frequency limit (Hz).

7.52.1 Detailed Description

Level: request.

Definition at line 387 of file Messages.hh.

7.52.2 Member Data Documentation

7.52.2.1 int _stLevelReq::iChannel

Channel.

Definition at line 389 of file Messages.hh.

Referenced by clLevel::ConnectStream(), clLevelMsg::GetRequest(), and clGUILevel::SendSettings().

7.52.2.2 int _stLevelReq::iAlgorithm

Algorithm.

Definition at line 390 of file Messages.hh.

Referenced by clLevelMsg::GetRequest(), clLevel::Initialize(), clLevel::Process(), and clGUILevel::SendSettings().

7.52.2.3 float _stLevelReq::fIntegrationTime

Itegration time (s).

Definition at line 391 of file Messages.hh.

Referenced by clGUILevel::BuildTable2(), clGUILevel::GetCfg(), clLevelMsg::GetRequest(), clLevel::Initialize(), and clGUILevel::SendSettings().

7.52.2.4 float _stLevelReq::fLowFrequency

Lower frequency limit (Hz).

Definition at line 392 of file Messages.hh.

Referenced by clLevelMsg::GetRequest(), clLevel::Initialize(), and clGUILevel::SendSettings().

7.52.2.5 float _stLevelReq::fHighFrequency

Higer frequency limit (Hz).

Definition at line 393 of file Messages.hh.

Referenced by clLevelMsg::GetRequest(), clLevel::Initialize(), and clGUILevel::SendSettings().

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

- **Messages.hh**

7.53 _stLevelRes Struct Reference

Level: result.

`#include <Messages.hh>`

Public Attributes

- timeval **sTimeStamp**

  Timestamp.

- float **fIntegrationTime**

  Integration time (s).

- float **fPeakLevel**

  Peak level (dB).

- float **fResult**

  Result.

7.53.1 Detailed Description

Level: result.

Definition at line 399 of file Messages.hh.

7.53.2 Member Data Documentation

7.53.2.1 struct timeval _stLevelRes::sTimeStamp

Timestamp.

Definition at line 401 of file Messages.hh.

Referenced by clLevelMsg::GetResult(), clLevel::Process(), and clLevelMsg::SetResult().

7.53.2.2 float _stLevelRes::fIntegrationTime

Integration time (s).

Definition at line 402 of file Messages.hh.

Referenced by clLevelMsg::GetResult(), clLevel::Initialize(), and clLevelMsg::SetResult().

7.53.2.3 float _stLevelRes::fPeakLevel

Peak level (dB).

Definition at line 403 of file Messages.hh.

Referenced by clLevelMsg::GetResult(), clLevel::SendResult(), and clLevelMsg::SetResult().

7.53.2.4 float _stLevelRes::fResult

Result.

Definition at line 404 of file Messages.hh.

Referenced by clLevelMsg::GetResult(), clLevel::Process(), and clLevelMsg::SetResult().

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

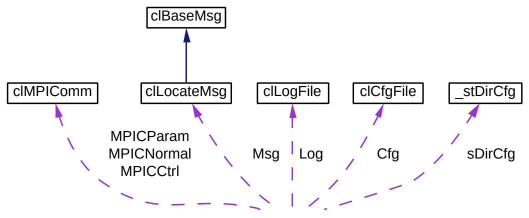
- **Messages.hh**

7.54 clLocate Class Reference

Master node for locate server.

`#include <Locate.hh>`

Collaboration diagram for clLocate:



Public Member Functions

- **clLocate** ()
- ∼**clLocate** ()
- int **Main** (int, char ∗∗)
- void **Stop** ()
- void ∗ **ProcessThread** (void ∗)
- void ∗ **ServeClientThread** (void ∗)

Private Member Functions

- bool **Initialize** ()
- void **SendParams** (int, const char ∗, int, long, long, float)

Private Attributes

- volatile bool **bRun**
- int **iSensorCount**
- int **iPort**
- int **iMsgSize**
- long **lWidth**
- long **lHeight**
- GDT **fWeight**
- GDT ∗ **fpLocMatrix**
- **stDirCfg sDirCfg**
- clAlloc **LocMatrix**
- clAlloc **ResMsg**
- clCondition **CndResMsg**
- clMutex **MtxResMsg**

- **clCfgFile Cfg**
- clDSPOp **DSP**
- **clLogFile Log**
- **clLocateMsg Msg**
- **clMPIComm MPICCtrl**
- **clMPIComm MPICParam**
- **clMPIComm MPICNormal**

7.54.1 Detailed Description

Master node for locate server.

Definition at line 73 of file Locate.hh.

7.54.2 Constructor & Destructor Documentation

7.54.2.1 clLocate::clLocate ()

Definition at line 220 of file Locate.cc.

References clLogFile::Add(), bRun, iSensorCount, LOCATE_CONTROL_TAG, LOCATE_LOGFILE, LOCATE_NORMAL_TAG, LOCATE_PARAM_TAG, Log, MPICCtrl, MPICNormal, MPICParam, clLogFile::Open(), and clMPIComm::SetTag().

7.54.2.2 clLocate::∼clLocate ()

Definition at line 232 of file Locate.cc.

References clLogFile::Add(), and Log.

7.54.3 Member Function Documentation

7.54.3.1 bool clLocate::Initialize () `[private]`

Definition at line 137 of file Locate.cc.

References Cfg, DSP, _stDirCfg::fAlpha, _stDirCfg::fIntegrationTime, _stDirCfg::fLowFrequency, fpLocMatrix, _stDirCfg::fScalingExp, _stDirCfg::fSoundSpeed, fWeight, GDT, clCfgFile::GetFlt(), clCfgFile::GetInt(), GLOBAL_HEADER_LEN, iMsgSize, iNodeCount, iPort, _stDirCfg::iRemoveNoise, _stDirCfg::iScaling, iSensorCount, _stDirCfg::lGapLength, lHeight, _stDirCfg::lMeanLength, LOCATE_CFGFILE, LOCATE_DEF_ALPHA, LOCATE_DEF_GAPLENGTH, LOCATE_DEF_INTTIME, LOCATE_DEF_LOWFREQ, LOCATE_DEF_MEANLENGTH, LOCATE_DEF_REMOVENOISE, LOCATE_DEF_SCALING, LOCATE_DEF_SCALINGEXP, LOCATE_DEF_SOUNDSPEED, LOCATE_DEF_WINDOWSIZE, LOCATE_HOSTNAME_MAXLEN, LOCATE_SENSOR_LIST, LocMatrix, lWidth, _stDirCfg::lWindowSize, ResMsg, sDirCfg, SendParams(), and clCfgFile::SetFileName().

Referenced by Main().

7.54.3.2 void clLocate::SendParams (int, const char *, int, long, long, float) [private]

Definition at line 200 of file Locate.cc.

References lHeight, LOCATE_HOSTNAME_MAXLEN, lWidth, MPICParam, sDirCfg, and clMPIComm::Send().

Referenced by Initialize().

7.54.3.3 int clLocate::Main (int, char **)

Definition at line 238 of file Locate.cc.

References bDebug, clSockServ::Bind(), bRun, Initialize(), iPort, LOCATE_TIMEOUT, MasterSigHandler(), clSockServ::WaitForConnect(), WrapProcessThread(), and WrapServeClientThread().

Referenced by main().

7.54.3.4 void clLocate::Stop ()

Definition at line 267 of file Locate.cc.

References bRun, iNodeCount, LOCATE_CTRL_STOP, MPICCtrl, and clMPIComm::Send().

Referenced by MasterSigHandler().

7.54.3.5 void * clLocate::ProcessThread (void *)

Definition at line 278 of file Locate.cc.

References clLocateSystem::Add(), bDebug, bRun, CndResMsg, fpLocMatrix, fWeight, GDT, clLocateSystem::GetResults(), clLocateSystem::Initialize(), iNodeCount, lHeight, _stLocateRes::lPointCount, lWidth, MPICNormal, Msg, MtxResMsg, clLocateSystem::Process(), clMPIComm::Recv(), ResMsg, and clLocateMsg::SetResult().

Referenced by WrapProcessThread().

7.54.3.6 void * clLocate::ServeClientThread (void *)

Definition at line 319 of file Locate.cc.

References bRun, CndResMsg, GLOBAL_HEADER_LEN, _stLocateHdr::iHeight, iMsgSize, _stLocateHdr::iWidth, lHeight, lWidth, Msg, MtxResMsg, ResMsg, clLocateMsg::SetHeader(), SOp, and clSockOp::WriteN().

Referenced by WrapServeClientThread().

7.54.4 Member Data Documentation

7.54.4.1 volatile bool clLocate::bRun [private]

Definition at line 75 of file Locate.hh.

Referenced by clLocate(), Main(), ProcessThread(), ServeClientThread(), and Stop().

7.54.4.2 int clLocate::iSensorCount [private]

Definition at line 76 of file Locate.hh.

Referenced by clLocate(), and Initialize().

7.54.4.3 int clLocate::iPort [private]

Definition at line 77 of file Locate.hh.

Referenced by Initialize(), and Main().

7.54.4.4 int clLocate::iMsgSize [private]

Definition at line 78 of file Locate.hh.

Referenced by Initialize(), and ServeClientThread().

7.54.4.5 long clLocate::lWidth [private]

Definition at line 79 of file Locate.hh.

Referenced by Initialize(), ProcessThread(), SendParams(), and ServeClientThread().

7.54.4.6 long clLocate::lHeight [private]

Definition at line 80 of file Locate.hh.

Referenced by Initialize(), ProcessThread(), SendParams(), and ServeClientThread().

7.54.4.7 GDT clLocate::fWeight [private]

Definition at line 81 of file Locate.hh.

Referenced by Initialize(), and ProcessThread().

7.54.4.8 GDT* clLocate::fpLocMatrix [private]

Definition at line 82 of file Locate.hh.

Referenced by Initialize(), and ProcessThread().

7.54.4.9 stDirCfg clLocate::sDirCfg [private]

Definition at line 83 of file Locate.hh.

Referenced by Initialize(), and SendParams().

7.54.4.10 clAlloc clLocate::LocMatrix [private]

Definition at line 84 of file Locate.hh.

Referenced by Initialize().

7.54.4.11 clAlloc clLocate::ResMsg [private]

Definition at line 85 of file Locate.hh.

Referenced by Initialize(), ProcessThread(), and ServeClientThread().

7.54.4.12 clCondition clLocate::CndResMsg [private]

Definition at line 86 of file Locate.hh.

Referenced by ProcessThread(), and ServeClientThread().

7.54.4.13 clMutex clLocate::MtxResMsg [private]

Definition at line 87 of file Locate.hh.

Referenced by ProcessThread(), and ServeClientThread().

7.54.4.14 clCfgFile clLocate::Cfg [private]

Definition at line 88 of file Locate.hh.

Referenced by Initialize().

7.54.4.15 clDSPOp clLocate::DSP [private]

Definition at line 89 of file Locate.hh.

Referenced by Initialize().

7.54.4.16 clLogFile clLocate::Log [private]

Definition at line 90 of file Locate.hh.

Referenced by clLocate(), and ~clLocate().

7.54.4.17 clLocateMsg clLocate::Msg [private]

Definition at line 91 of file Locate.hh.

Referenced by ProcessThread(), and ServeClientThread().

7.54.4.18 clMPIComm clLocate::MPICCtrl [private]

Definition at line 92 of file Locate.hh.

Referenced by clLocate(), and Stop().

7.54.4.19 clMPIComm clLocate::MPICParam [private]

Definition at line 93 of file Locate.hh.

Referenced by clLocate(), and SendParams().

7.54.4.20 clMPIComm clLocate::MPICNormal [private]

Definition at line 94 of file Locate.hh.

Referenced by clLocate(), and ProcessThread().

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

- **Locate.hh**
- **Locate.cc**

7.55 _stLocateHdr Struct Reference

Locate: results header.

`#include <Messages.hh>`

Public Attributes

- int **iWidth**
  Matrix width.
- int **iHeight**
  Matrix height.

7.55.1 Detailed Description

Locate: results header.

Definition at line 410 of file Messages.hh.

7.55.2 Member Data Documentation

7.55.2.1 int _stLocateHdr::iWidth

Matrix width.

Definition at line 412 of file Messages.hh.

Referenced by clGUILocate::DisplayResults(), clLocateMsg::GetHeader(), main(), clGUILocate::OnConnectClick(), clGUILocate::OnLocateExpose(), clLocate::ServeClientThread(), and clLocateMsg::SetHeader().

7.55.2.2 int _stLocateHdr::iHeight

Matrix height.

Definition at line 413 of file Messages.hh.

Referenced by clGUILocate::DisplayResults(), clLocateMsg::GetHeader(), main(), clGUILocate::OnConnectClick(), clGUILocate::OnLocateExpose(), clLocate::ServeClientThread(), and clLocateMsg::SetHeader().

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

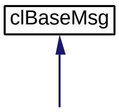
- **Messages.hh**

7.56 clLocateMsg Class Reference

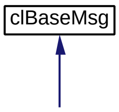
Locate server communication.

`#include <Messages.hh>`

Inheritance diagram for clLocateMsg:



Collaboration diagram for clLocateMsg:



Public Member Functions

- void **SetHeader** (char ∗, const **stpLocateHdr**)
- void **GetHeader** (const char ∗, **stpLocateHdr**)
- void **SetResult** (void ∗, const **stpLocateRes**, const float ∗)
- void **SetResult** (void ∗, const **stpLocateRes**, const double ∗)
- void **GetResult** (const void ∗, **stpLocateRes**, float ∗)
- void **GetResult** (const void ∗, **stpLocateRes**, double ∗)

7.56.1 Detailed Description

Locate server communication.

Definition at line 603 of file Messages.hh.

7.56.2 Member Function Documentation

7.56.2.1 void clLocateMsg::SetHeader (char ∗, const *stpLocateHdr*)

Definition at line 1289 of file Messages.cc.

References GLOBAL_HEADER_LEN, _stLocateHdr::iHeight, and _stLocateHdr::iWidth.

Referenced by clLocate::ServeClientThread().

7.56.2.2 void clLocateMsg::GetHeader (const char *, stpLocateHdr)

Definition at line 1304 of file Messages.cc.

References _stLocateHdr::iHeight, and _stLocateHdr::iWidth.

Referenced by clGUILocate::ConnectToServer(), and main().

7.56.2.3 void clLocateMsg::SetResult (void *, const *stpLocateRes*, const float *)

Definition at line 1312 of file Messages.cc.

References clBaseMsg::EndianConv(), GLOBAL_HEADER_LEN, and _stLocateRes::lPointCount.

Referenced by clLocate::ProcessThread().

7.56.2.4 void clLocateMsg::SetResult (void *, const *stpLocateRes*, const double *)

Definition at line 1335 of file Messages.cc.

References clBaseMsg::EndianConv(), GLOBAL_HEADER_LEN, and _stLocateRes::lPointCount.

7.56.2.5 void clLocateMsg::GetResult (const void *, stpLocateRes, float *)

Definition at line 1358 of file Messages.cc.

References clBaseMsg::EndianConv(), GLOBAL_HEADER_LEN, and _stLocateRes::lPointCount.

Referenced by main(), and clGUILocate::OnGdkInput().

7.56.2.6 void clLocateMsg::GetResult (const void *, stpLocateRes, double *)

Definition at line 1375 of file Messages.cc.

References clBaseMsg::EndianConv(), GLOBAL_HEADER_LEN, and _stLocateRes::lPointCount.

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

- **Messages.hh**
- **Messages.cc**

7.57 _stLocateRes Struct Reference

Locate: result.

`#include <Messages.hh>`

Public Attributes

- long **lPointCount**
  Point count.

7.57.1 Detailed Description

Locate: result.

Definition at line 419 of file Messages.hh.

7.57.2 Member Data Documentation

7.57.2.1 long _stLocateRes::lPointCount

Point count.

Definition at line 421 of file Messages.hh.

Referenced by clGUILocate::DisplayResults(), clLocateMsg::GetResult(), clLocate::ProcessThread(), and clLocateMsg::SetResult().

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

- **Messages.hh**

7.58 clLocateSensor Class Reference

Sensor matrix processing for locating sound sources.

`#include <LocateSensor.hh>`

Public Member Functions

- **clLocateSensor** ()
- ~**clLocateSensor** ()
- bool **Initialize** (long, long, long, long, GDT, bool)

  Initialize.

- void **Uninitialize** ()

  Uninitialize.

- void **SetDirectionValues** (const GDT *, const GDT *, long, long, long, GDT)

  Set direction values.

- GDT * **GetResultMatrix** ()

  Get result matrix.

Private Member Functions

- void **Clear** ()
- void **SetValue** (long, long, GDT)
- void **SetDirection** (GDT, GDT, GDT)

Private Attributes

- bool **bInitialized**
- bool **b3D**
- long **lWidth**
- long **lHeight**
- long **lPosX**
- long **lPosY**
- GDT **fPI**
- GDT **fHalfPI**
- GDT **fAzimuth**
- GDT * **fpLocMatrix**
- clAlloc **LocMatrix**
- clDSPOp **DSP**

7.58.1 Detailed Description

Sensor matrix processing for locating sound sources.

Definition at line 36 of file LocateSensor.hh.

7.58.2 Constructor & Destructor Documentation

7.58.2.1 clLocateSensor::clLocateSensor ()

Definition at line 83 of file LocateSensor.cc.

References bInitialized, fHalfPI, and fPI.

7.58.2.2 clLocateSensor::~clLocateSensor ()

Definition at line 91 of file LocateSensor.cc.

References bInitialized, and Uninitialize().

7.58.3 Member Function Documentation

7.58.3.1 void clLocateSensor::Clear () `[inline, private]`

Definition at line 31 of file LocateSensor.cc.

References fpLocMatrix, GDT, lHeight, and lWidth.

Referenced by Initialize(), and SetDirectionValues().

7.58.3.2 void clLocateSensor::SetValue (long, long, GDT) `[inline, private]`

Definition at line 37 of file LocateSensor.cc.

References fpLocMatrix, lHeight, and lWidth.

Referenced by SetDirection().

7.58.3.3 void clLocateSensor::SetDirection (GDT, GDT, GDT) `[private]`

Definition at line 45 of file LocateSensor.cc.

References b3D, DSP, fAzimuth, fHalfPI, fPI, GDT, lHeight, lPosX, lPosY, and SetValue().

Referenced by SetDirectionValues().

7.58.3.4 bool clLocateSensor::Initialize (long, long, long, long, GDT, bool)

Initialize.

Parameters:

lW Matrix width

lH Matrix height

lX Sensor X position

lY Sensor Y position

fA Sensor azimuth (direction) (rad)

bTD Three dimensional array (non-twosided case)

Returns:

Success

Definition at line 97 of file LocateSensor.cc.

References b3D, Clear(), fAzimuth, fpLocMatrix, GDT, lHeight, LocMatrix, lPosX, lPosY, and lWidth.

Referenced by clSubLocate::Initialize().

7.58.3.5 void clLocateSensor::Uninitialize ()

Uninitialize.

Definition at line 119 of file LocateSensor.cc.

Referenced by ∼clLocateSensor().

7.58.3.6 void clLocateSensor::SetDirectionValues (const GDT *, const GDT *, long, long, long, GDT)

Set direction values.

We just walk through matrix and adjust level using

$$\alpha = 0.05 \frac{f}{1000}^{1.4}$$

$$R = 15 \log(s) + \alpha s 10^{-3}$$

$$V_n = V_0 10^{\frac{R}{20}}$$

We definitely need way better algorithm, this is stupidly simple. Ajust multiplier for R for environment, 10 gives cylindral spreading and 20 spherical spreading.

Parameters:

fpLevelValues Levels

fpDirectionValues Directions
lValueCount number of levels & directions
lMinBin Results starting at index
lMaxBin Results end at index
fFreqResolution Frequency resolution

Definition at line 125 of file LocateSensor.cc.

References Clear(), GDT, and SetDirection().

Referenced by clSubLocate::Main().

7.58.3.7 GDT* clLocateSensor::GetResultMatrix () `[inline]`

Get result matrix.

Returns:
Pointer to result matrix

Definition at line 99 of file LocateSensor.hh.

References fpLocMatrix, and GDT.

Referenced by clSubLocate::Main().

7.58.4 Member Data Documentation

7.58.4.1 bool clLocateSensor::bInitialized `[private]`

Definition at line 38 of file LocateSensor.hh.

Referenced by clLocateSensor(), and ~clLocateSensor().

7.58.4.2 bool clLocateSensor::b3D `[private]`

Definition at line 39 of file LocateSensor.hh.

Referenced by Initialize(), and SetDirection().

7.58.4.3 long clLocateSensor::lWidth `[private]`

Definition at line 40 of file LocateSensor.hh.

Referenced by Clear(), Initialize(), and SetValue().

7.58.4.4 long clLocateSensor::lHeight `[private]`

Definition at line 41 of file LocateSensor.hh.

Referenced by Clear(), Initialize(), SetDirection(), and SetValue().

7.58.4.5 long clLocateSensor::lPosX [private]

Definition at line 42 of file LocateSensor.hh.

Referenced by Initialize(), and SetDirection().

7.58.4.6 long clLocateSensor::lPosY [private]

Definition at line 43 of file LocateSensor.hh.

Referenced by Initialize(), and SetDirection().

7.58.4.7 GDT clLocateSensor::fPI [private]

Definition at line 44 of file LocateSensor.hh.

Referenced by clLocateSensor(), and SetDirection().

7.58.4.8 GDT clLocateSensor::fHalfPI [private]

Definition at line 45 of file LocateSensor.hh.

Referenced by clLocateSensor(), and SetDirection().

7.58.4.9 GDT clLocateSensor::fAzimuth [private]

Definition at line 46 of file LocateSensor.hh.

Referenced by Initialize(), and SetDirection().

7.58.4.10 GDT* clLocateSensor::fpLocMatrix [private]

Definition at line 47 of file LocateSensor.hh.

Referenced by Clear(), GetResultMatrix(), Initialize(), and SetValue().

7.58.4.11 clAlloc clLocateSensor::LocMatrix [private]

Definition at line 48 of file LocateSensor.hh.

Referenced by Initialize().

7.58.4.12 clDSPOp clLocateSensor::DSP [private]

Definition at line 49 of file LocateSensor.hh.

Referenced by SetDirection().

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

- **LocateSensor.hh**
- **LocateSensor.cc**

7.59 clLocateSystem Class Reference

Result combining of locate matrixes.

`#include <LocateSystem.hh>`

Public Member Functions

- **clLocateSystem** ()
- ~**clLocateSystem** ()
- bool **Initialize** (long, long, GDT)

  Initialize.
- void **Add** (const GDT *)

  Add subresult.
- void **Process** ()

  Process final result.
- void **GetResults** (GDT *)

  Get results.

Private Attributes

- long **lWidth**
- long **lHeight**
- long **lPointCount**
- long **lResultCount**
- GDT **fWeight**
- GDT * **fpResults**
- GDT * **fpFinal**
- clAlloc **Results**
- clAlloc **Final**
- clDSPOp **DSP**

7.59.1 Detailed Description

Result combining of locate matrixes.

Definition at line 36 of file LocateSystem.hh.

7.59.2 Constructor & Destructor Documentation

7.59.2.1 clLocateSystem::clLocateSystem ()

Definition at line 29 of file LocateSystem.cc.

7.59.2.2 clLocateSystem::~clLocateSystem ()

Definition at line 34 of file LocateSystem.cc.

7.59.3 Member Function Documentation

7.59.3.1 bool clLocateSystem::Initialize (long, long, GDT)

Initialize.

Parameters:

lW Matrix width

lH Matrix height

fCoeff Weighting coefficient

Definition at line 39 of file LocateSystem.cc.

References DSP, Final, fpFinal, fpResults, fWeight, GDT, lHeight, lPointCount, lResultCount, lWidth, and Results.

Referenced by clLocate::ProcessThread().

7.59.3.2 void clLocateSystem::Add (const GDT *)

Add subresult.

Parameters:

fpSubRes Result matrix

Definition at line 61 of file LocateSystem.cc.

References DSP, fpResults, lPointCount, and lResultCount.

Referenced by clLocate::ProcessThread().

7.59.3.3 void clLocateSystem::Process ()

Process final result.

Definition at line 68 of file LocateSystem.cc.

References DSP, fpFinal, fpResults, fWeight, GDT, lPointCount, and lResultCount.

Referenced by clLocate::ProcessThread().

7.59.3.4 void clLocateSystem::GetResults (GDT *)

Get results.

Parameters:
fpDest Result matrix

Definition at line 82 of file LocateSystem.cc.

References DSP, fpFinal, and lPointCount.

Referenced by clLocate::ProcessThread().

7.59.4 Member Data Documentation

7.59.4.1 long clLocateSystem::lWidth `[private]`

Definition at line 38 of file LocateSystem.hh.

Referenced by Initialize().

7.59.4.2 long clLocateSystem::lHeight `[private]`

Definition at line 39 of file LocateSystem.hh.

Referenced by Initialize().

7.59.4.3 long clLocateSystem::lPointCount `[private]`

Definition at line 40 of file LocateSystem.hh.

Referenced by Add(), GetResults(), Initialize(), and Process().

7.59.4.4 long clLocateSystem::lResultCount `[private]`

Definition at line 41 of file LocateSystem.hh.

Referenced by Add(), Initialize(), and Process().

7.59.4.5 GDT clLocateSystem::fWeight `[private]`

Definition at line 42 of file LocateSystem.hh.

Referenced by Initialize(), and Process().

7.59.4.6 GDT* clLocateSystem::fpResults `[private]`

Definition at line 43 of file LocateSystem.hh.

Referenced by Add(), Initialize(), and Process().

7.59.4.7 GDT* clLocateSystem::fpFinal [private]

Definition at line 44 of file LocateSystem.hh.

Referenced by GetResults(), Initialize(), and Process().

7.59.4.8 clAlloc clLocateSystem::Results [private]

Definition at line 45 of file LocateSystem.hh.

Referenced by Initialize().

7.59.4.9 clAlloc clLocateSystem::Final [private]

Definition at line 46 of file LocateSystem.hh.

Referenced by Initialize().

7.59.4.10 clDSPOp clLocateSystem::DSP [private]

Definition at line 47 of file LocateSystem.hh.

Referenced by Add(), GetResults(), Initialize(), and Process().

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

- **LocateSystem.hh**
- **LocateSystem.cc**

7.60 _stLofarCursor Struct Reference

Lofar cursor.

`#include <GUILofar.hh>`

Public Attributes

- int **iType**

  Cursor type.

- int **iPosition**

  Position.

- int **iDistance**

  Spike spacing.

7.60.1 Detailed Description

Lofar cursor.

Definition at line 108 of file GUILofar.hh.

7.60.2 Member Data Documentation

7.60.2.1 int _stLofarCursor::iType

Cursor type.

Definition at line 110 of file GUILofar.hh.

Referenced by clGUILofar::DrawCursor(), and clGUILofar::OnMotionLofar().

7.60.2.2 int _stLofarCursor::iPosition

Position.

Definition at line 111 of file GUILofar.hh.

Referenced by clGUILofar::DrawCursor(), clGUILofar::OnMotionLofar(), and clGUILofar::PrintStatus().

7.60.2.3 int _stLofarCursor::iDistance

Spike spacing.

Definition at line 112 of file GUILofar.hh.

Referenced by clGUILofar::DrawCursor(), clGUILofar::OnMotionLofar(), and clGUILofar::PrintStatus().

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

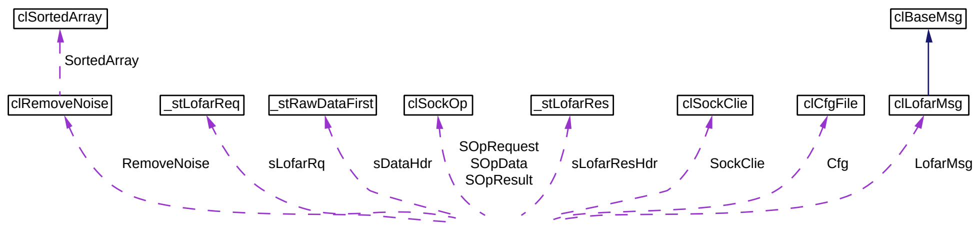
- **GUILofar.hh**

7.61 clLofarDemon Class Reference

LOFAR/DEMON calculation server.

`#include <LofarDemon.hh>`

Collaboration diagram for clLofarDemon:



Public Member Functions

- **clLofarDemon** (int, int)
- ∼**clLofarDemon** ()
- int **Exec** ()

Public Attributes

- volatile bool **bRun**

Private Member Functions

- bool **GetRequestMsg** ()
- bool **GetDataHdr** ()
- bool **Initialize** ()
- bool **InitFilter** ()
- bool **InitFFT** ()
- int **MainLoop** ()
- void **CalcSpect** ()
- bool **SendResults** ()
- void **PutData** (GDT ∗, long)
- bool **PullData** (GDT ∗, long)
- void **DemonProc** (GDT ∗, long)
- void **SpectStdDev** (GDT ∗)

Private Attributes

- bool **bReverseOrder**
- int **iFilterType**
- int **iDCBlock**
- int **iNormIdx**
- int **iNormHistCount**
- long **lFilterSize**
- long **lDecimation**
- long **lDemonDecimation**
- long **lWinSize**
- long **lSpectSize**
- long **lNewDataSize**
- long **lOldDataSize**
- long **lAvgCntr**
- long **lHistCntr**
- GDT * **fpWinFunc**
- GDT * **fpDataBuf**
- GDT * **fpSpectRes**
- GDT * **fpLofarRes**
- GDT * **fpNormHist**
- **stRawDataFirst sDataHdr**
- **stLofarReq sLofarRq**
- **stLofarRes sLofarResHdr**
- clDSPAlloc **WinFunc**
- clDSPAlloc **DataBuf**
- clDSPAlloc **SpectRes**
- clDSPAlloc **LofarRes**
- clDSPAlloc **HistoryBuf**
- clDSPAlloc **NormHist**
- **clCfgFile Cfg**
- clDSPOp **DSP**
- **clLofarMsg LofarMsg**
- clReBuffer **ReBuffer**

  This is used instead of decimator if decimation factor is < 2.

- clRecDecimator **Decimator**

  This is always the last decimation.

- clRecDecimator **DemonDecimator**

  DEMON BP decimation.

- clIIRCascade **DCBlockIIR**

  DC notch for DEMON.

- clFlipBand **FlipBand**

Odd frequency band flipping.

- **clRemoveNoise RemoveNoise**
- **clSockClie SockClie**
- **clSockOp SOpRequest**
- **clSockOp SOpResult**
- **clSockOp SOpData**

7.61.1 Detailed Description

LOFAR/DEMON calculation server.

Definition at line 47 of file LofarDemon.hh.

7.61.2 Constructor & Destructor Documentation

7.61.2.1 clLofarDemon::clLofarDemon (int, int)

Definition at line 758 of file LofarDemon.cc.

References bRun, fpDataBuf, _stLofarRes::fPeakLevel, fpLofarRes, fpSpectRes, fpWinFunc, lAvgCntr, lHistCntr, clSockOp::SetHandle(), sLofarResHdr, SOpRequest, and SOpResult.

7.61.2.2 clLofarDemon::~clLofarDemon ()

Definition at line 773 of file LofarDemon.cc.

7.61.3 Member Function Documentation

7.61.3.1 bool clLofarDemon::GetRequestMsg () `[private]`

Definition at line 104 of file LofarDemon.cc.

References clLofarMsg::GetRequest(), GLOBAL_HEADER_LEN, LOFAR_REQ_TIMEOUT, LofarMsg, clSockOp::ReadN(), clSockOp::ReadSelect(), sLofarRq, and SOpRequest.

Referenced by Exec().

7.61.3.2 bool clLofarDemon::GetDataHdr () `[private]`

Definition at line 117 of file LofarDemon.cc.

References Cfg, clSockClie::Connect(), clCfgFile::GetStr(), _stLofarReq::iChannel, _stRawDataReq::iChannel, LOFAR_RAW1ST_TIMEOUT, clSockOp::ReadN(), clSockOp::ReadSelect(), SD_CFGFILE, sDataHdr, clCfgFile::SetFileName(), clSockOp::SetHandle(), sLofarRq, SockClie, SOpData, and clSockOp::WriteN().

Referenced by Exec().

7.61.3.3 bool clLofarDemon::Initialize () [private]

Definition at line 141 of file LofarDemon.cc.

References Cfg, InitFFT(), InitFilter(), LOFAR_CFGFILE, and clCfgFile::SetFileName().

Referenced by Exec().

7.61.3.4 bool clLofarDemon::InitFilter () [private]

Definition at line 150 of file LofarDemon.cc.

References bDebug, _stLofarReq::bDemon, bReverseOrder, Cfg, DCBlockIIR, Decimator, DemonDecimator, _stRawDataFirst::dSampleRate, _stLofarRes::fDemonBand, _stLofarRes::fHighFreq, _stLofarReq::fHighFreq, FlipBand, _stLofarRes::fLowFreq, _stLofarReq::fLowFreq, GDT, clCfgFile::GetInt(), iDCBlock, iFilterType, lDecimation, lDemonDecimation, lFilterSize, LOFAR_DEF_DC_BLOCK, LOFAR_DEF_FILTER_SIZE, sDataHdr, sLofarResHdr, and sLofarRq.

Referenced by Initialize().

7.61.3.5 bool clLofarDemon::InitFFT () [private]

Definition at line 255 of file LofarDemon.cc.

References bDebug, Cfg, cpaWindowFuncs, DataBuf, DSP, fpDataBuf, fpLofarRes, fpNormHist, fpSpectRes, fpWinFunc, _stLofarReq::fWinParameter, GDT, clCfgFile::GetInt(), HistoryBuf, _stLofarReq::iClip, iNormHistCount, iNormIdx, _stLofarReq::iOverlap, _stLofarReq::iType, _stLofarReq::iWindow, _stLofarReq::lMeanLength, lNewDataSize, LOFAR_NWINS, LofarRes, lOldDataSize, _stLofarRes::lSpectLength, lSpectSize, _stLofarReq::lWinLength, lWinSize, MSG_LOFAR_CLIP_SLIDING, MSG_LOFAR_TYPE_AUTOCORR, MSG_LOFAR_TYPE_CEPSTRUM, MSG_LOFAR_TYPE_FFT, MSG_LOFAR_WIN_BARTLETT, MSG_LOFAR_WIN_BLACKMAN, MSG_LOFAR_WIN_BM_HARRIS, MSG_LOFAR_WIN_COS_TAPER, MSG_LOFAR_WIN_EXP, MSG_LOFAR_WIN_FLATTOP, MSG_LOFAR_WIN_HAMMING, MSG_LOFAR_WIN_HANNING, MSG_LOFAR_WIN_KAI_BES, MSG_LOFAR_WIN_KAISER, MSG_LOFAR_WIN_RECT, MSG_LOFAR_WIN_TUKEY, NormHist, sLofarResHdr, sLofarRq, SpectRes, and WinFunc.

Referenced by Initialize().

7.61.3.6 int clLofarDemon::MainLoop () [private]

Definition at line 348 of file LofarDemon.cc.

References bDebug, _stLofarReq::bDemon, bRun, CalcSpect(), _stRawDataFirst::dSampleRate, DSP, _stLofarRes::fLineTime, fpDataBuf, _stLofarRes::fPeakLevel, fpLofarRes, GDT, clSockOp::GetRecvBufSize(), lAvgCntr, _stLofarReq::lAvgCount, lDecimation, lDemonDecimation, lFilterSize, lNewDataSize, lOldDataSize, lSpectSize, _stLofarReq::lWinLength, PullData(), PutData(), clSockOp::ReadN(), sDataHdr, SendResults(), clSockOp::SetRecvBufSize(), sLofarResHdr, sLofarRq, SOpData, and _stLofarRes::sTimeStamp.

Referenced by Exec().

7.61.3.7 void clLofarDemon::CalcSpect () [private]

Definition at line 417 of file LofarDemon.cc.

References _stLofarReq::bDemon, _stLofarReq::bLinear, clRemoveNoise::Diff(), DSP, _stLofarReq::fAlpha, _stLofarRes::fLowFreq, fpDataBuf, fpLofarRes, fpNormHist, fpSpectRes, fpWinFunc, GCDT, GDT, _stLofarReq::iClip, iNormHistCount, iNormIdx, clRemoveNoise::InvDiff(), _stLofarReq::iRemoveNoise, _stLofarReq::iType, lAvgCntr, _stLofarReq::lAvgCount, _stLofarReq::lGapLength, _stLofarReq::lMeanLength, lSpectSize, _stLofarReq::lWinLength, lWinSize, MSG_LOFAR_BNER_DIFF, MSG_LOFAR_BNER_IDIFF, MSG_LOFAR_BNER_NONE, MSG_LOFAR_BNER_OTA, MSG_LOFAR_BNER_STDDEV, MSG_LOFAR_BNER_TPSW, MSG_LOFAR_CLIP_10DB, MSG_LOFAR_CLIP_20DB, MSG_LOFAR_CLIP_50P, MSG_LOFAR_CLIP_75P, MSG_LOFAR_CLIP_BOTH, MSG_LOFAR_CLIP_LOW, MSG_LOFAR_CLIP_MEAN, MSG_LOFAR_CLIP_MEDIAN, MSG_LOFAR_CLIP_NONE, MSG_LOFAR_CLIP_OFFSET, MSG_LOFAR_CLIP_OFFSET2, MSG_LOFAR_CLIP_OFFSET3, MSG_LOFAR_CLIP_SLIDING, MSG_LOFAR_TYPE_AUTOCORR, MSG_LOFAR_TYPE_CEPSTRUM, MSG_LOFAR_TYPE_FFT, clRemoveNoise::OTA(), RemoveNoise, sLofarResHdr, sLofarRq, SpectStdDev(), and clRemoveNoise::TPSW().

Referenced by MainLoop().

7.61.3.8 bool clLofarDemon::SendResults () [private]

Definition at line 614 of file LofarDemon.cc.

References bDebug, fpLofarRes, GDT, clSockOp::GetErrno(), GLOBAL_HEADER_LEN, LOFAR_TIMEOUT, LofarMsg, lSpectSize, clLofarMsg::SetResult(), sLofarResHdr, SOpResult, clSockOp::WriteN(), and clSockOp::WriteSelect().

Referenced by MainLoop().

7.61.3.9 void clLofarDemon::PutData (GDT *, long) [private]

Definition at line 644 of file LofarDemon.cc.

References _stLofarReq::bDemon, bReverseOrder, DCBlockIIR, Decimator, DemonDecimator, DSP, FlipBand, iDCBlock, lDecimation, ReBuffer, and sLofarRq.

Referenced by MainLoop().

7.61.3.10 bool clLofarDemon::PullData (GDT *, long) [private]

Definition at line 694 of file LofarDemon.cc.

References _stLofarReq::bDemon, bReverseOrder, Decimator, FlipBand, lDecimation, ReBuffer, and sLofarRq.

Referenced by MainLoop().

7.61.3.11 void clLofarDemon::DemonProc (GDT *, long) [private]

Definition at line 718 of file LofarDemon.cc.

7.61.3.12 void clLofarDemon::SpectStdDev (GDT *) [private]

Definition at line 729 of file LofarDemon.cc.

References DSP, GDT, HistoryBuf, lHistCntr, _stLofarReq::lMeanLength, lSpectSize, and sLofarRq.

Referenced by CalcSpect().

7.61.3.13 int clLofarDemon::Exec ()

Definition at line 778 of file LofarDemon.cc.

References bDebug, GetDataHdr(), GetRequestMsg(), Initialize(), and MainLoop().

Referenced by main().

7.61.4 Member Data Documentation

7.61.4.1 bool clLofarDemon::bReverseOrder [private]

Definition at line 49 of file LofarDemon.hh.

Referenced by InitFilter(), PullData(), and PutData().

7.61.4.2 int clLofarDemon::iFilterType [private]

Definition at line 50 of file LofarDemon.hh.

Referenced by InitFilter().

7.61.4.3 int clLofarDemon::iDCBlock [private]

Definition at line 51 of file LofarDemon.hh.

Referenced by InitFilter(), and PutData().

7.61.4.4 int clLofarDemon::iNormIdx [private]

Definition at line 52 of file LofarDemon.hh.

Referenced by CalcSpect(), and InitFFT().

7.61.4.5 int clLofarDemon::iNormHistCount [private]

Definition at line 53 of file LofarDemon.hh.

Referenced by CalcSpect(), and InitFFT().

7.61.4.6 long clLofarDemon::lFilterSize [private]

Definition at line 54 of file LofarDemon.hh.

Referenced by InitFilter(), and MainLoop().

7.61.4.7 long clLofarDemon::lDecimation [private]

Definition at line 55 of file LofarDemon.hh.

Referenced by InitFilter(), MainLoop(), PullData(), and PutData().

7.61.4.8 long clLofarDemon::lDemonDecimation [private]

Definition at line 56 of file LofarDemon.hh.

Referenced by InitFilter(), and MainLoop().

7.61.4.9 long clLofarDemon::lWinSize [private]

Definition at line 57 of file LofarDemon.hh.

Referenced by CalcSpect(), and InitFFT().

7.61.4.10 long clLofarDemon::lSpectSize [private]

Definition at line 58 of file LofarDemon.hh.

Referenced by CalcSpect(), InitFFT(), MainLoop(), SendResults(), and SpectStdDev().

7.61.4.11 long clLofarDemon::lNewDataSize [private]

Definition at line 59 of file LofarDemon.hh.

Referenced by InitFFT(), and MainLoop().

7.61.4.12 long clLofarDemon::lOldDataSize [private]

Definition at line 60 of file LofarDemon.hh.

Referenced by InitFFT(), and MainLoop().

7.61.4.13 long clLofarDemon::lAvgCntr [private]

Definition at line 61 of file LofarDemon.hh.

Referenced by CalcSpect(), clLofarDemon(), and MainLoop().

7.61.4.14 long clLofarDemon::lHistCntr [private]

Definition at line 62 of file LofarDemon.hh.

Referenced by clLofarDemon(), and SpectStdDev().

7.61.4.15 GDT* clLofarDemon::fpWinFunc [private]

Definition at line 63 of file LofarDemon.hh.

Referenced by CalcSpect(), clLofarDemon(), and InitFFT().

7.61.4.16 GDT* clLofarDemon::fpDataBuf [private]

Definition at line 64 of file LofarDemon.hh.

Referenced by CalcSpect(), clLofarDemon(), InitFFT(), and MainLoop().

7.61.4.17 GDT* clLofarDemon::fpSpectRes [private]

Definition at line 65 of file LofarDemon.hh.

Referenced by CalcSpect(), clLofarDemon(), and InitFFT().

7.61.4.18 GDT* clLofarDemon::fpLofarRes [private]

Definition at line 66 of file LofarDemon.hh.

Referenced by CalcSpect(), clLofarDemon(), InitFFT(), MainLoop(), and SendResults().

7.61.4.19 GDT* clLofarDemon::fpNormHist [private]

Definition at line 67 of file LofarDemon.hh.

Referenced by CalcSpect(), and InitFFT().

7.61.4.20 stRawDataFirst clLofarDemon::sDataHdr [private]

Definition at line 68 of file LofarDemon.hh.

Referenced by GetDataHdr(), InitFilter(), and MainLoop().

7.61.4.21 stLofarReq clLofarDemon::sLofarRq [private]

Definition at line 69 of file LofarDemon.hh.

Referenced by CalcSpect(), GetDataHdr(), GetRequestMsg(), InitFFT(), InitFilter(), MainLoop(), PullData(), PutData(), and SpectStdDev().

7.61.4.22 stLofarRes clLofarDemon::sLofarResHdr [private]

Definition at line 70 of file LofarDemon.hh.

Referenced by CalcSpect(), clLofarDemon(), InitFFT(), InitFilter(), MainLoop(), and SendResults().

7.61.4.23 clDSPAlloc clLofarDemon::WinFunc [private]

Definition at line 71 of file LofarDemon.hh.

Referenced by InitFFT().

7.61.4.24 clDSPAlloc clLofarDemon::DataBuf [private]

Definition at line 72 of file LofarDemon.hh.

Referenced by InitFFT().

7.61.4.25 clDSPAlloc clLofarDemon::SpectRes [private]

Definition at line 73 of file LofarDemon.hh.

Referenced by InitFFT().

7.61.4.26 clDSPAlloc clLofarDemon::LofarRes [private]

Definition at line 74 of file LofarDemon.hh.

Referenced by InitFFT().

7.61.4.27 clDSPAlloc clLofarDemon::HistoryBuf [private]

Definition at line 75 of file LofarDemon.hh.

Referenced by InitFFT(), and SpectStdDev().

7.61.4.28 clDSPAlloc clLofarDemon::NormHist [private]

Definition at line 76 of file LofarDemon.hh.

Referenced by InitFFT().

7.61.4.29 clCfgFile clLofarDemon::Cfg [private]

Definition at line 77 of file LofarDemon.hh.

Referenced by GetDataHdr(), InitFFT(), InitFilter(), and Initialize().

7.61.4.30 clDSPOp clLofarDemon::DSP [private]

Definition at line 78 of file LofarDemon.hh.

Referenced by CalcSpect(), InitFFT(), MainLoop(), PutData(), and SpectStdDev().

7.61.4.31 clLofarMsg clLofarDemon::LofarMsg [private]

Definition at line 79 of file LofarDemon.hh.

Referenced by GetRequestMsg(), and SendResults().

7.61.4.32 clReBuffer clLofarDemon::ReBuffer [private]

This is used instead of decimator if decimation factor is < 2.

Definition at line 80 of file LofarDemon.hh.

Referenced by PullData(), and PutData().

7.61.4.33 clRecDecimator clLofarDemon::Decimator [private]

This is always the last decimation.

Definition at line 81 of file LofarDemon.hh.

Referenced by InitFilter(), PullData(), and PutData().

7.61.4.34 clRecDecimator clLofarDemon::DemonDecimator [private]

DEMON BP decimation.

Definition at line 82 of file LofarDemon.hh.

Referenced by InitFilter(), and PutData().

7.61.4.35 clIIRCascade clLofarDemon::DCBlockIIR [private]

DC notch for DEMON.

Definition at line 83 of file LofarDemon.hh.

Referenced by InitFilter(), and PutData().

7.61.4.36 clFlipBand clLofarDemon::FlipBand [private]

Odd frequency band flipping.

Definition at line 84 of file LofarDemon.hh.

Referenced by InitFilter(), PullData(), and PutData().

7.61.4.37 clRemoveNoise clLofarDemon::RemoveNoise [private]

Definition at line 85 of file LofarDemon.hh.

Referenced by CalcSpect().

7.61.4.38 clSockClie clLofarDemon::SockClie [private]

Definition at line 86 of file LofarDemon.hh.

Referenced by GetDataHdr().

7.61.4.39 clSockOp clLofarDemon::SOpRequest [private]

Definition at line 87 of file LofarDemon.hh.

Referenced by clLofarDemon(), and GetRequestMsg().

7.61.4.40 clSockOp clLofarDemon::SOpResult [private]

Definition at line 88 of file LofarDemon.hh.

Referenced by clLofarDemon(), and SendResults().

7.61.4.41 clSockOp clLofarDemon::SOpData [private]

Definition at line 89 of file LofarDemon.hh.

Referenced by GetDataHdr(), and MainLoop().

7.61.4.42 volatile bool clLofarDemon::bRun

Definition at line 103 of file LofarDemon.hh.

Referenced by clLofarDemon(), and MainLoop().

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

- **LofarDemon.hh**
- **LofarDemon.cc**

7.62 _stLOFARInfo Struct Reference

Information stored in saved LOFAR .tif.inf.

`#include <ConvPic.hh>`

Public Attributes

- time_t **ttTime**

  Oldest dataline time.

- double **dLowFreq**

  Lower frequency limit.

- double **dHighFreq**

  Upper frequency limit.

- double **dLineTime**

  Length of one scanline in time (s).

7.62.1 Detailed Description

Information stored in saved LOFAR .tif.inf.

Definition at line 62 of file ConvPic.hh.

7.62.2 Member Data Documentation

7.62.2.1 time_t _stLOFARInfo::ttTime

Oldest dataline time.

Definition at line 64 of file ConvPic.hh.

Referenced by clConvPic::DrawLOFARInfo(), and clConvPic::LoadLOFARInfo().

7.62.2.2 double _stLOFARInfo::dLowFreq

Lower frequency limit.

Definition at line 65 of file ConvPic.hh.

Referenced by clConvPic::DrawLOFARInfo(), and clConvPic::LoadLOFARInfo().

7.62.2.3 double _stLOFARInfo::dHighFreq

Upper frequency limit.

Definition at line 66 of file ConvPic.hh.

Referenced by clConvPic::DrawLOFARInfo(), and clConvPic::LoadLOFARInfo().

7.62.2.4 double _stLOFARInfo::dLineTime

Length of one scanline in time (s).

Definition at line 67 of file ConvPic.hh.

Referenced by clConvPic::DrawLOFARInfo(), and clConvPic::LoadLOFARInfo().

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

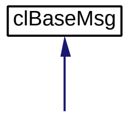
- **ConvPic.hh**

7.63 clLofarMsg Class Reference

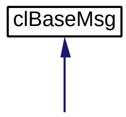
LOFAR/DEMON server communication.

#include <Messages.hh>

Inheritance diagram for clLofarMsg:



Collaboration diagram for clLofarMsg:



Public Member Functions

- void **SetRequest** (char *, const **stpLofarReq**)
- void **GetRequest** (const char *, **stpLofarReq**)
- void **SetResult** (void *, const **stpLofarRes**, const float *)
- void **SetResult** (void *, const **stpLofarRes**, const double *)
- void **GetResult** (const void *, **stpLofarRes**, float *)
- void **GetResult** (const void *, **stpLofarRes**, double *)

7.63.1 Detailed Description

LOFAR/DEMON server communication.

Definition at line 551 of file Messages.hh.

7.63.2 Member Function Documentation

7.63.2.1 void clLofarMsg::SetRequest (char *, const *stpLofarReq*)

Definition at line 854 of file Messages.cc.

References _stLofarReq::bDemon, _stLofarReq::bLinear, _stLofarReq::fAlpha, _stLofarReq::fHighFreq, _stLofarReq::fLowFreq, _stLofarReq::fWinParameter, GLOBAL_HEADER_LEN, _stLofarReq::iChannel, _stLofarReq::iClip, _stLofarReq::iOverlap, _stLofarReq::iRemoveNoise, _stLofarReq::iType, _stLofarReq::iWindow, _stLofarReq::lAvgCount, _stLofarReq::lGapLength, _stLofarReq::lMeanLength, _stLofarReq::lWinLength, MSG_FALSE, and MSG_TRUE.

Referenced by clGUILofar::SendSettings().

7.63.2.2 void clLofarMsg::GetRequest (const char *, stpLofarReq)

Definition at line 903 of file Messages.cc.

References _stLofarReq::bDemon, _stLofarReq::bLinear, _stLofarReq::fAlpha, _stLofarReq::fHighFreq, _stLofarReq::fLowFreq, _stLofarReq::fWinParameter, _stLofarReq::iChannel, _stLofarReq::iClip, _stLofarReq::iOverlap, _stLofarReq::iRemoveNoise, _stLofarReq::iType, _stLofarReq::iWindow, _stLofarReq::lAvgCount, _stLofarReq::lGapLength, _stLofarReq::lMeanLength, _stLofarReq::lWinLength, and MSG_TRUE.

Referenced by clLofarDemon::GetRequestMsg().

7.63.2.3 void clLofarMsg::SetResult (void *, const *stpLofarRes*, const float *)

Definition at line 930 of file Messages.cc.

References clBaseMsg::EndianConv(), _stLofarRes::fDemonBand, _stLofarRes::fHighFreq, _stLofarRes::fLineTime, _stLofarRes::fLowFreq, _stLofarRes::fPeakLevel, GLOBAL_HEADER_LEN, _stLofarRes::lSpectLength, and _stLofarRes::sTimeStamp.

Referenced by clLofarDemon::SendResults().

7.63.2.4 void clLofarMsg::SetResult (void *, const *stpLofarRes*, const double *)

Definition at line 966 of file Messages.cc.

References clBaseMsg::EndianConv(), _stLofarRes::fDemonBand, _stLofarRes::fHighFreq, _stLofarRes::fLineTime, _stLofarRes::fLowFreq, _stLofarRes::fPeakLevel, GLOBAL_HEADER_LEN, _stLofarRes::lSpectLength, and _stLofarRes::sTimeStamp.

7.63.2.5 void clLofarMsg::GetResult (const void *, stpLofarRes, float *)

Definition at line 1002 of file Messages.cc.

References clBaseMsg::EndianConv(), _stLofarRes::fDemonBand, _stLofarRes::fHighFreq, _stLofarRes::fLineTime, _stLofarRes::fLowFreq, _stLofarRes::fPeakLevel, GLOBAL_HEADER_LEN, _stLofarRes::lSpectLength, and _stLofarRes::sTimeStamp.

Referenced by clGUILofar::OnGdkInput().

7.63.2.6 void clLofarMsg::GetResult (const void *, stpLofarRes, double *)

Definition at line 1029 of file Messages.cc.

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

- **Messages.hh**
- **Messages.cc**

7.64 _stLofarReq Struct Reference

LOFAR: request.

`#include <Messages.hh>`

Public Attributes

- int **iChannel**

  Channel.

- int **iType**

  Type; FFT, Hankel, Cepstrum, etc.

- int **iWindow**

  Window function.

- float **fWinParameter**

  Parameter for window function.

- long **lWinLength**

  Window length in points.

- float **fLowFreq**

  Lower frequency limit (Hz).

- float **fHighFreq**

  Higher frequency limit (Hz).

- int **iOverlap**

  Overlap ().

- bool **bLinear**

  Linear/logarithmic scaling.

- int **iRemoveNoise**

  Background noise estimation and removal algorithm.

- float **fAlpha**

  Alpha for noise removal.

- long **lMeanLength**

  Mean length for noise removal.

- long **lGapLength**

  Gap length for TPSW noise removal algorithm.

- bool **bDemon**

  LOFAR/DEMON.

- int **iClip**

  Predefined clipping function.

- long **lAvgCount**

  Number of spectrums to average.

7.64.1 Detailed Description

LOFAR: request.

Definition at line 320 of file Messages.hh.

7.64.2 Member Data Documentation

7.64.2.1 int _stLofarReq::iChannel

Channel.

Definition at line 322 of file Messages.hh.

Referenced by clLofarDemon::GetDataHdr(), clLofarMsg::GetRequest(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.2 int _stLofarReq::iType

Type; FFT, Hankel, Cepstrum, etc.

Definition at line 323 of file Messages.hh.

Referenced by clLofarDemon::CalcSpect(), clLofarMsg::GetRequest(), clLofarDemon::InitFFT(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.3 int _stLofarReq::iWindow

Window function.

Definition at line 324 of file Messages.hh.

Referenced by clLofarMsg::GetRequest(), clLofarDemon::InitFFT(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.4 float _stLofarReq::fWinParameter

Parameter for window function.

Definition at line 325 of file Messages.hh.

Referenced by clLofarMsg::GetRequest(), clLofarDemon::InitFFT(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.5 long _stLofarReq::lWinLength

Window length in points.

Definition at line 326 of file Messages.hh.

Referenced by clGUILofar::BuildTable2(), clGUILofar::BuildTableLofar(), clLofarDemon::CalcSpect(), clGUILofar::clGUILofar(), clLofarMsg::GetRequest(), clGUILofar::InitConnection(), clLofarDemon::InitFFT(), clLofarDemon::MainLoop(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.6 float _stLofarReq::fLowFreq

Lower frequency limit (Hz).

Definition at line 327 of file Messages.hh.

Referenced by clGUILofar::BuildTable2(), clGUILofar::clGUILofar(), clLofarMsg::GetRequest(), clLofarDemon::InitFilter(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.7 float _stLofarReq::fHighFreq

Higher frequency limit (Hz).

Definition at line 328 of file Messages.hh.

Referenced by clGUILofar::BuildTable2(), clGUILofar::clGUILofar(), clLofarMsg::GetRequest(), clLofarDemon::InitFilter(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.8 int _stLofarReq::iOverlap

Overlap ().

Definition at line 329 of file Messages.hh.

Referenced by clLofarMsg::GetRequest(), clLofarDemon::InitFFT(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.9 bool _stLofarReq::bLinear

Linear/logarithmic scaling.

Definition at line 330 of file Messages.hh.

Referenced by clLofarDemon::CalcSpect(), clLofarMsg::GetRequest(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.10 int _stLofarReq::iRemoveNoise

Background noise estimation and removal algorithm.

Definition at line 331 of file Messages.hh.

Referenced by clGUILofar::BuildTable3(), clLofarDemon::CalcSpect(), clGUILofar::clGUILofar(), clLofarMsg::GetRequest(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.11 float _stLofarReq::fAlpha

Alpha for noise removal.

Definition at line 332 of file Messages.hh.

Referenced by clGUILofar::BuildTable3(), clLofarDemon::CalcSpect(), clGUILofar::clGUILofar(), clLofarMsg::GetRequest(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.12 long _stLofarReq::lMeanLength

Mean length for noise removal.

Definition at line 333 of file Messages.hh.

Referenced by clGUILofar::BuildTable3(), clLofarDemon::CalcSpect(), clGUILofar::clGUILofar(), clLofarMsg::GetRequest(), clLofarDemon::InitFFT(), clGUILofar::SendSettings(), clLofarMsg::SetRequest(), and clLofarDemon::SpectStdDev().

7.64.2.13 long _stLofarReq::lGapLength

Gap length for TPSW noise removal algorithm.

Definition at line 334 of file Messages.hh.

Referenced by clGUILofar::BuildTable3(), clLofarDemon::CalcSpect(), clGUILofar::clGUILofar(), clLofarMsg::GetRequest(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.14 bool _stLofarReq::bDemon

LOFAR/DEMON.

Definition at line 335 of file Messages.hh.

Referenced by clLofarDemon::CalcSpect(), clGUILofar::Configure(), clLofarMsg::GetRequest(), clLofarDemon::InitFilter(), clLofarDemon::MainLoop(), clGUILofar::OnSaveClicks(), clGUILofar::PrintStatus(), clLofarDemon::PullData(), clLofarDemon::PutData(), clGUILofar::SaveInfo(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.15 int _stLofarReq::iClip

Predefined clipping function.

Definition at line 336 of file Messages.hh.

Referenced by clGUILofar::BuildTable3(), clLofarDemon::CalcSpect(), clGUILofar::clGUILofar(), clLofarMsg::GetRequest(), clLofarDemon::InitFFT(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

7.64.2.16 long _stLofarReq::lAvgCount

Number of spectrums to average.

Definition at line 337 of file Messages.hh.

Referenced by clGUILofar::BuildTable3(), clLofarDemon::CalcSpect(), clGUILofar::clGUILofar(), clLofarMsg::GetRequest(), clLofarDemon::MainLoop(), clGUILofar::SendSettings(), and clLofarMsg::SetRequest().

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

- **Messages.hh**

7.65 _stLofarRes Struct Reference

LOFAR: result.

`#include <Messages.hh>`

Public Attributes

- timeval **sTimeStamp**

  Timestamp.

- long **lSpectLength**

  Spectrum length.

- float **fLowFreq**

  Lower frequency limit (Hz).

- float **fHighFreq**

  Higher frequency limit (Hz).

- float **fDemonBand**

  DEMON bandwidth (Hz).

- float **fLineTime**

  Length of spectrum window in time (s).

- float **fPeakLevel**

  Peak level (dB).

7.65.1 Detailed Description

LOFAR: result.

Definition at line 343 of file Messages.hh.

7.65.2 Member Data Documentation

7.65.2.1 struct timeval _stLofarRes::sTimeStamp

Timestamp.

Definition at line 345 of file Messages.hh.

Referenced by clLofarMsg::GetResult(), clLofarDemon::MainLoop(), clGUILofar::OnGdkInput(), clGUILofar::PrintStatus(), and clLofarMsg::SetResult().

7.65.2.2 long _stLofarRes::lSpectLength

Spectrum length.

Definition at line 346 of file Messages.hh.

Referenced by clGUILofar::Configure(), clLofarMsg::GetResult(), clLofarDemon::InitFFT(), clGUILofar::PrintStatus(), and clLofarMsg::SetResult().

7.65.2.3 float _stLofarRes::fLowFreq

Lower frequency limit (Hz).

Definition at line 347 of file Messages.hh.

Referenced by clLofarDemon::CalcSpect(), clGUILofar::Configure(), clLofarMsg::GetResult(), clLofarDemon::InitFilter(), clGUILofar::OnSaveClicks(), clGUILofar::PrintStatus(), clGUILofar::SaveInfo(), and clLofarMsg::SetResult().

7.65.2.4 float _stLofarRes::fHighFreq

Higher frequency limit (Hz).

Definition at line 348 of file Messages.hh.

Referenced by clGUILofar::Configure(), clLofarMsg::GetResult(), clLofarDemon::InitFilter(), clGUILofar::OnSaveClicks(), clGUILofar::PrintStatus(), clGUILofar::SaveInfo(), and clLofarMsg::SetResult().

7.65.2.5 float _stLofarRes::fDemonBand

DEMON bandwidth (Hz).

Definition at line 349 of file Messages.hh.

Referenced by clGUILofar::Configure(), clLofarMsg::GetResult(), clLofarDemon::InitFilter(), clGUILofar::OnSaveClicks(), clGUILofar::PrintStatus(), clGUILofar::SaveInfo(), and clLofarMsg::SetResult().

7.65.2.6 float _stLofarRes::fLineTime

Length of spectrum window in time (s).

Definition at line 350 of file Messages.hh.

Referenced by clGUILofar::Configure(), clLofarMsg::GetResult(), clLofarDemon::MainLoop(), clGUILofar::OnGdkInput(), clGUILofar::OnSaveClicks(), clGUILofar::PrintStatus(), clGUILofar::SaveInfo(), and clLofarMsg::SetResult().

7.65.2.7 float _stLofarRes::fPeakLevel

Peak level (dB).

Definition at line 351 of file Messages.hh.

Referenced by clLofarDemon::clLofarDemon(), clLofarMsg::GetResult(), clLofarDemon::MainLoop(), clGUILofar::OnGdkInput(), clGUILofar::PrintStatus(), and clLofarMsg::SetResult().

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

- **Messages.hh**

7.66 clLogFile Class Reference

Log file class.

`#include <LogFile.hh>`

Public Member Functions

- **clLogFile** ()
- **clLogFile** (const char ∗)
- ∼**clLogFile** ()
- bool **Open** (const char ∗)

  Open logfile.

- bool **Add** (char, const char ∗)

  Add new entry to logfile.

- bool **Add** (char, const char ∗, int)

  Add new entry to logfile with errno.

- int **GetError** ()

  Get error code.

Private Member Functions

- void **Time** ()

Private Attributes

- bool **bOk**
- int **iEC**
- char **cpTime** [20]
- FILE ∗ **FLog**

7.66.1 Detailed Description

Log file class.

Definition at line 35 of file LogFile.hh.

7.66.2 Constructor & Destructor Documentation

7.66.2.1 clLogFile::clLogFile ()

Definition at line 31 of file LogFile.cc.

References bOk, iEC, and LOGFILE_NOERROR.

7.66.2.2 clLogFile::clLogFile (const char *)

Definition at line 38 of file LogFile.cc.

References bOk, iEC, LOGFILE_NOERROR, and Open().

7.66.2.3 clLogFile::~clLogFile ()

Definition at line 46 of file LogFile.cc.

References bOk, and FLog.

7.66.3 Member Function Documentation

7.66.3.1 void clLogFile::Time () `[private]`

Definition at line 96 of file LogFile.cc.

References cpTime.

Referenced by Add().

7.66.3.2 bool clLogFile::Open (const char *)

Open logfile.

Parameters:

cpLogFile Filename

Returns:

Success

Definition at line 52 of file LogFile.cc.

References bOk, FLog, and iEC.

Referenced by clBeamSrv2Master::clBeamSrv2Master(), clBeamSrvMaster::clBeamSrvMaster(), clComediSrv::clComediSrv(), clLocate::clLocate(), clLogFile(), clSaveSrv::clSaveSrv(), clSoundProxy::clSoundProxy(), clSoundSrv2::clSoundSrv2(), clSoundSrvA::clSoundSrvA(), and clStreamDist::clStreamDist().

7.66.3.3 bool clLogFile::Add (char, const char *)

Add new entry to logfile.

Parameters:

cLogMark Mark character

cpLogEntry Entry line

Returns:
Success

Definition at line 69 of file LogFile.cc.

References bOk, cpTime, FLog, and Time().

Referenced by clSoundProxy::AddToLog(), clStreamDist::AudioInThread(), clSoundSrvA::AudioInThread(), clSoundSrv2::AudioInThread(), clComediSrv::AudioInThread(), clBeamSrv2Master::clBeamSrv2Master(), clBeamSrvMaster::clBeamSrvMaster(), clComediSrv::clComediSrv(), clLocate::clLocate(), clSaveSrv::clSaveSrv(), clSoundSrv::clSoundSrv(), clSoundSrv2::clSoundSrv2(), clSoundSrvA::clSoundSrvA(), clStreamDist::clStreamDist(), clUIServer::clUIServer(), clSaveSrv::ConnectStream(), clSaveSrv::CreateFile(), clSaveSrv::CreateFile2(), clSoundSrv::Exec(), clSoundSrvA::GetAudioCfg(), clSoundSrv2::GetAudioCfg(), clComediSrv::GetAudioCfg(), clStreamDist::InitCompress(), clSoundSrvA::InitCompress(), clSoundSrv2::InitCompress(), clComediSrv::InitCompress(), clBeamSrv2Master::InitFilterBank(), clBeamSrvMaster::InitFilterBank(), clSaveSrv::Initialize(), clBeamSrv2Master::InitProcessing(), clBeamSrvMaster::InitProcessing(), clSoundSrv::InputExec(), clStreamDist::Main(), clSoundSrvA::Main(), clSoundSrv2::Main(), clComediSrv::Main(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clSaveSrv::ProcessLoop(), clBeamSrv2Master::ProcessLoop(), clBeamSrvMaster::ProcessLoop(), clSoundSrv::Quit(), clBeamSrv2Master::ReadConfig(), clBeamSrvMaster::ReadConfig(), clBeamSrvMaster::SendInData(), clBeamSrvMaster::SendNodeParams(), clSoundSrv::ServeClientExec(), clStreamDist::ServeClientThread(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clComediSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), SigHandler(), clUIServer::Wait(), clSoundSrv::WaitConnectExec(), clBeamSrv2Master::WaitReady(), clBeamSrvMaster::WaitReady(), clSaveSrv::WriteData(), clBeamSrv2Master::~clBeamSrv2Master(), clBeamSrvMaster::~clBeamSrvMaster(), clComediSrv::~clComediSrv(), clLocate::~clLocate(), clSaveSrv::~clSaveSrv(), clSoundSrv::~clSoundSrv(), clSoundSrv2::~clSoundSrv2(), clSoundSrvA::~clSoundSrvA(), clStreamDist::~clStreamDist(), and clUIServer::~clUIServer().

7.66.3.4 bool clLogFile::Add (char, const char *, int)

Add new entry to logfile with errno.

Parameters:
cLogMark Mark character
cpLogEntry Entry line
iErrno errno value

Returns:
Success

Definition at line 82 of file LogFile.cc.

References bOk, cpTime, FLog, and Time().

7.66.3.5 int clLogFile::GetError () [inline]

Get error code.

Definition at line 73 of file LogFile.hh.

References iEC.

Referenced by clUIServer::clUIServer().

7.66.4 Member Data Documentation

7.66.4.1 bool clLogFile::bOk [private]

Definition at line 37 of file LogFile.hh.

Referenced by Add(), clLogFile(), Open(), and ~clLogFile().

7.66.4.2 int clLogFile::iEC [private]

Definition at line 38 of file LogFile.hh.

Referenced by clLogFile(), GetError(), and Open().

7.66.4.3 char clLogFile::cpTime[20] [private]

Definition at line 39 of file LogFile.hh.

Referenced by Add(), and Time().

7.66.4.4 FILE* clLogFile::FLog [private]

Definition at line 40 of file LogFile.hh.

Referenced by Add(), Open(), and ~clLogFile().

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

- **LogFile.hh**
- **LogFile.cc**

7.67 clMPIComm Class Reference

MPI communication.

`#include <Cluster.hh>`

Public Member Functions

- **clMPIComm** ()
- **clMPIComm** (int)
- ∼**clMPIComm** ()
- void **SetTag** (int)

  Set message tag.

- bool **Send** (int, char ∗, int)

  Send message.

- bool **Send** (int, unsigned char ∗, int)
- bool **Send** (int, short ∗, int)
- bool **Send** (int, unsigned short ∗, int)
- bool **Send** (int, int ∗, int)
- bool **Send** (int, unsigned int ∗, int)
- bool **Send** (int, long ∗, int)
- bool **Send** (int, unsigned long ∗, int)
- bool **Send** (int, float ∗, int)
- bool **Send** (int, double ∗, int)
- bool **Send** (int, long double ∗, int)
- bool **Send** (int, void ∗, int)
- bool **Probe** (int)

  Synchronize message and return info (with our tag or wildcard).

- bool **ProbeAny** (int)
- bool **ProbeNB** (int)
- bool **ProbeAnyNB** (int)
- bool **Recv** (int, char ∗, int)

  Receive message.

- bool **Recv** (int, unsigned char ∗, int)
- bool **Recv** (int, short ∗, int)
- bool **Recv** (int, unsigned short ∗, int)
- bool **Recv** (int, int ∗, int)
- bool **Recv** (int, unsigned int ∗, int)
- bool **Recv** (int, long ∗, int)
- bool **Recv** (int, unsigned long ∗, int)
- bool **Recv** (int, float ∗, int)
- bool **Recv** (int, double ∗, int)

- bool **Recv** (int, long double ∗, int)
- bool **Recv** (int, void ∗, int)
- bool **RecvAny** (int, char ∗, int)

  Receive message with any tag.

- bool **RecvAny** (int, unsigned char ∗, int)
- bool **RecvAny** (int, short ∗, int)
- bool **RecvAny** (int, unsigned short ∗, int)
- bool **RecvAny** (int, int ∗, int)
- bool **RecvAny** (int, unsigned int ∗, int)
- bool **RecvAny** (int, long ∗, int)
- bool **RecvAny** (int, unsigned long ∗, int)
- bool **RecvAny** (int, float ∗, int)
- bool **RecvAny** (int, double ∗, int)
- bool **RecvAny** (int, long double ∗, int)
- bool **RecvAny** (int, void ∗, int)
- bool **GetCount** (const std::type_info &, int ∗)

  Get length of last received message (number of elements).

- int **GetSenderRank** ()

  Get rank of sender of last received message.

- int **GetSenderTag** ()

  Get tag of last received message.

- int **GetError** ()

  Get MPI returned error code.

- int **GetError2** ()

  Get MPI error value from status.

Private Attributes

- int **iTag**
- int **iError**
- MPI_Status **sStatus**

7.67.1 Detailed Description

MPI communication.

Definition at line 82 of file Cluster.hh.

7.67.2 Constructor & Destructor Documentation

7.67.2.1 clMPIComm::clMPIComm ()

Definition at line 81 of file Cluster.cc.

References iError, and iTag.

7.67.2.2 clMPIComm::clMPIComm (int)

Definition at line 88 of file Cluster.cc.

References iError, and iTag.

7.67.2.3 clMPIComm::~clMPIComm ()

Definition at line 95 of file Cluster.cc.

7.67.3 Member Function Documentation

7.67.3.1 void clMPIComm::SetTag (int)

Set message tag.

Parameters:
iTag Tag

Definition at line 100 of file Cluster.cc.

References iTag.

Referenced by clLocate::clLocate(), and clSubLocate::clSubLocate().

7.67.3.2 bool clMPIComm::Send (int, char *, int)

Send message.

Parameters:
iDest Receiver rank
Data Message buffer
iCount Item count

Returns:
Success

Definition at line 106 of file Cluster.cc.

References iError, and iTag.

Referenced by clSubLocate::Main(), clLocate::SendParams(), and clLocate::Stop().

7.67.3.3 bool clMPIComm::Send (int, unsigned char *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 115 of file Cluster.cc.

References iError, and iTag.

7.67.3.4 bool clMPIComm::Send (int, short *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 124 of file Cluster.cc.

References iError, and iTag.

7.67.3.5 bool clMPIComm::Send (int, unsigned short *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 133 of file Cluster.cc.

References iError, and iTag.

7.67.3.6 bool clMPIComm::Send (int, int *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 142 of file Cluster.cc.

References iError, and iTag.

7.67.3.7 bool clMPIComm::Send (int, unsigned int *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 151 of file Cluster.cc.

References iError, and iTag.

7.67.3.8 bool clMPIComm::Send (int, long *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 160 of file Cluster.cc.

References iError, and iTag.

7.67.3.9 bool clMPIComm::Send (int, unsigned long *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 169 of file Cluster.cc.

References iError, and iTag.

7.67.3.10 bool clMPIComm::Send (int, float *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 178 of file Cluster.cc.

References iError, and iTag.

7.67.3.11 bool clMPIComm::Send (int, double *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 187 of file Cluster.cc.

References iError, and iTag.

7.67.3.12 bool clMPIComm::Send (int, long double *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 196 of file Cluster.cc.

References iError, and iTag.

7.67.3.13 bool clMPIComm::Send (int, void *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 205 of file Cluster.cc.

References iError, and iTag.

7.67.3.14 bool clMPIComm::Probe (int)

Synchronize message and return info (with our tag or wildcard).

Check/wait for message.

Parameters:

iSrc Source rank

Returns:
Status

Definition at line 214 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.15 bool clMPIComm::ProbeAny (int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 222 of file Cluster.cc.

References iError, and sStatus.

Referenced by clSubLocate::RecvParams().

7.67.3.16 bool clMPIComm::ProbeNB (int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 230 of file Cluster.cc.

References iError, iTag, and sStatus.

Referenced by clSubLocate::Main().

7.67.3.17 bool clMPIComm::ProbeAnyNB (int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 241 of file Cluster.cc.

References iError, and sStatus.

7.67.3.18 bool clMPIComm::Recv (int, char *, int)

Receive message.

Parameters:
iSrc Source rank
Data Message buffer
iCount Buffer size in items

Returns:
Success

Definition at line 252 of file Cluster.cc.

References iError, iTag, and sStatus.

Referenced by clSubLocate::Main(), clLocate::ProcessThread(), and clSubLocate::RecvParams().

7.67.3.19 bool clMPIComm::Recv (int, unsigned char *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 261 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.20 bool clMPIComm::Recv (int, short *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 270 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.21 bool clMPIComm::Recv (int, unsigned short *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 279 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.22 bool clMPIComm::Recv (int, int *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 288 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.23 bool clMPIComm::Recv (int, unsigned int *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 297 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.24 bool clMPIComm::Recv (int, long *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 306 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.25 bool clMPIComm::Recv (int, unsigned long *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 315 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.26 bool clMPIComm::Recv (int, float *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 324 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.27 bool clMPIComm::Recv (int, double *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 333 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.28 bool clMPIComm::Recv (int, long double *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 342 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.29 bool clMPIComm::Recv (int, void *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 351 of file Cluster.cc.

References iError, iTag, and sStatus.

7.67.3.30 bool clMPIComm::RecvAny (int, char *, int)

Receive message with any tag.

Parameters:

iSrc Sender rank

Data Message buffer

iCount Size of message buffer in items

Definition at line 360 of file Cluster.cc.

References iError, and sStatus.

7.67.3.31 bool clMPIComm::RecvAny (int, unsigned char *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 369 of file Cluster.cc.

References iError, and sStatus.

7.67.3.32 bool clMPIComm::RecvAny (int, short *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 378 of file Cluster.cc.

References iError, and sStatus.

7.67.3.33 bool clMPIComm::RecvAny (int, unsigned short *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 387 of file Cluster.cc.

References iError, and sStatus.

7.67.3.34 bool clMPIComm::RecvAny (int, int *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 396 of file Cluster.cc.

References iError, and sStatus.

7.67.3.35 bool clMPIComm::RecvAny (int, unsigned int *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 405 of file Cluster.cc.

References iError, and sStatus.

7.67.3.36 bool clMPIComm::RecvAny (int, long *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 414 of file Cluster.cc.

References iError, and sStatus.

7.67.3.37 bool clMPIComm::RecvAny (int, unsigned long *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 423 of file Cluster.cc.

References iError, and sStatus.

7.67.3.38 bool clMPIComm::RecvAny (int, float *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 432 of file Cluster.cc.

References iError, and sStatus.

7.67.3.39 bool clMPIComm::RecvAny (int, double *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 441 of file Cluster.cc.

References iError, and sStatus.

7.67.3.40 bool clMPIComm::RecvAny (int, long double *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 450 of file Cluster.cc.

References iError, and sStatus.

7.67.3.41 bool clMPIComm::RecvAny (int, void *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 459 of file Cluster.cc.

References iError, and sStatus.

7.67.3.42 bool clMPIComm::GetCount (const std::type_info &, int *)

Get length of last received message (number of elements).

Note:
This is not to be used extensively. This uses RTTI, so it could be very slow.

Parameters:
TypeInfo Type information
ipCount Number of elements

Returns:
Success

Definition at line 468 of file Cluster.cc.

References iError, and sStatus.

7.67.3.43 int clMPIComm::GetSenderRank () `[inline]`

Get rank of sender of last received message.

Returns:
Sender rank

Definition at line 220 of file Cluster.hh.

References sStatus.

7.67.3.44 int clMPIComm::GetSenderTag () `[inline]`

Get tag of last received message.

Returns:
Last received tag

Definition at line 226 of file Cluster.hh.

References sStatus.

Referenced by clSubLocate::RecvParams().

7.67.3.45 int clMPIComm::GetError () [inline]

Get MPI returned error code.

Returns:
Error code

Definition at line 232 of file Cluster.hh.

References iError.

7.67.3.46 int clMPIComm::GetError2 () [inline]

Get MPI error value from status.

Returns:
Error code

Definition at line 238 of file Cluster.hh.

References sStatus.

7.67.4 Member Data Documentation

7.67.4.1 int clMPIComm::iTag [private]

Definition at line 84 of file Cluster.hh.

Referenced by clMPIComm(), Probe(), ProbeNB(), Recv(), Send(), and SetTag().

7.67.4.2 int clMPIComm::iError [private]

Definition at line 85 of file Cluster.hh.

Referenced by clMPIComm(), GetCount(), GetError(), Probe(), ProbeAny(), ProbeAnyNB(), ProbeNB(), Recv(), RecvAny(), and Send().

7.67.4.3 MPI_Status clMPIComm::sStatus [private]

Definition at line 86 of file Cluster.hh.

Referenced by GetCount(), GetError2(), GetSenderRank(), GetSenderTag(), Probe(), ProbeAny(), ProbeAnyNB(), ProbeNB(), Recv(), and RecvAny().

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

- **Cluster.hh**
- **Cluster.cc**

7.68 clMPIProc Class Reference

MPI process.

`#include <Cluster.hh>`

Public Member Functions

- **clMPIProc** ()
- ~**clMPIProc** ()
- bool **Initialize** (int ∗, char ∗∗∗)

  Initialize MPI process.

- bool **Finalize** ()

  Uninitialize MPI process.

- bool **GetRank** (int ∗)

  Get rank of this MPI node.

- bool **GetNodeCount** (int ∗)

  Get number of nodes for this MPI process.

- int **GetError** ()

  Get error code.

Private Attributes

- bool **bInitialized**
- int **iError**

7.68.1 Detailed Description

MPI process.

Definition at line 35 of file Cluster.hh.

7.68.2 Constructor & Destructor Documentation

7.68.2.1 clMPIProc::clMPIProc ()

Definition at line 30 of file Cluster.cc.

References bInitialized, and iError.

7.68.2.2 clMPIProc::~clMPIProc ()

Definition at line 37 of file Cluster.cc.

References bInitialized, and Finalize().

7.68.3 Member Function Documentation

7.68.3.1 bool clMPIProc::Initialize (int *, char ***)

Initialize MPI process.

Parameters:

argc Parameter count

argv Parameter vector

Returns:

Success

Definition at line 43 of file Cluster.cc.

References bInitialized, and iError.

Referenced by main().

7.68.3.2 bool clMPIProc::Finalize ()

Uninitialize MPI process.

Returns:

Success

Definition at line 53 of file Cluster.cc.

References bInitialized, and iError.

Referenced by main(), and ~clMPIProc().

7.68.3.3 bool clMPIProc::GetRank (int *)

Get rank of this MPI node.

Parameters:

ipRank Rank

Returns:

Success

Definition at line 63 of file Cluster.cc.

References bInitialized, and iError.

Referenced by main().

7.68.3.4 bool clMPIProc::GetNodeCount (int *)

Get number of nodes for this MPI process.

Parameters:
ipNodeCount Number of nodes

Returns:
Success

Definition at line 72 of file Cluster.cc.

References bInitialized, and iError.

Referenced by main().

7.68.3.5 int clMPIProc::GetError () `[inline]`

Get error code.

Returns:
Error code

Definition at line 75 of file Cluster.hh.

References iError.

7.68.4 Member Data Documentation

7.68.4.1 bool clMPIProc::bInitialized `[private]`

Definition at line 37 of file Cluster.hh.

Referenced by clMPIProc(), Finalize(), GetNodeCount(), GetRank(), Initialize(), and ∼clMPIProc().

7.68.4.2 int clMPIProc::iError `[private]`

Definition at line 38 of file Cluster.hh.

Referenced by clMPIProc(), Finalize(), GetError(), GetNodeCount(), GetRank(), and Initialize().

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

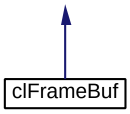
- **Cluster.hh**
- **Cluster.cc**

7.69 clPalette Class Reference

RGB palette class.

#include <Palette.hh>

Inheritance diagram for clPalette:



Public Member Functions

- **clPalette** ()
- ~**clPalette** ()
- void **GenBW** ()

  Generate white-to-black palette.

- void **GenHSV** ()

  Generate black-blue-cyan-green-yellow-red palette.

- void **GenLight** ()

  Generate palette matching to light's spectrum.

- void **GenTemp** ()

  Generate black-red-yellow-white palette.

- void **GenDir** ()

  Generate black-white palette with two highest values red.

- void **GenGreen** ()

  Generate black-green-red palette.

- void **GenGreen2** ()

  Generate black-green-white palette.

- void **GenGreen3** ()

  Generate black-green-yellow palette.

- void **GenGreen4** ()

  Generate black-green-red-yellow palette.

- void **GenPureGreen** ()

Generate black-green palette.

- void **GenWB** ()

  Generate black-white palette.

- int **Size** ()

  Get number of colors in palette.

- unsigned int **Color** (int iValue)

  Return color from palette LUT.

- unsigned int **operator**[ ] (int iValue)

  Return color from palette LUT.

Protected Attributes

- int **iPalSize**
- unsigned int * **upPalette**

Private Attributes

- clAlloc **Palette**

7.69.1 Detailed Description

RGB palette class.

Definition at line 33 of file Palette.hh.

7.69.2 Constructor & Destructor Documentation

7.69.2.1 clPalette::clPalette ()

Definition at line 26 of file Palette.cc.

References iPalSize, and upPalette.

7.69.2.2 clPalette::~clPalette ()

Definition at line 33 of file Palette.cc.

7.69.3 Member Function Documentation

7.69.3.1 void clPalette::GenBW ()

Generate white-to-black palette.

Definition at line 38 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clGUILocate::GetCfg(), clGUILocate::OnPaletteActivate(), and clFrameBuf::PalGenBW().

7.69.3.2 void clPalette::GenHSV ()

Generate black-blue-cyan-green-yellow-red palette.

Definition at line 54 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clGUILocate::OnPaletteActivate(), and clFrameBuf::PalGenHSV().

7.69.3.3 void clPalette::GenLight ()

Generate palette matching to light's spectrum.

Definition at line 85 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clGUILocate::OnPaletteActivate(), and clFrameBuf::PalGenLight().

7.69.3.4 void clPalette::GenTemp ()

Generate black-red-yellow-white palette.

Definition at line 112 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clGUILocate::OnPaletteActivate(), and clFrameBuf::PalGenTemp().

7.69.3.5 void clPalette::GenDir ()

Generate black-white palette with two highest values red.

Definition at line 135 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clGUILocate::OnPaletteActivate(), and clFrameBuf::PalGenDir().

7.69.3.6 void clPalette::GenGreen ()

Generate black-green-red palette.

"NATO-style"

Definition at line 153 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clGUILocate::OnPaletteActivate(), and clFrameBuf::PalGenGreen().

7.69.3.7 void clPalette::GenGreen2 ()

Generate black-green-white palette.

Definition at line 172 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clGUILocate::OnPaletteActivate(), and clFrameBuf::PalGenGreen2().

7.69.3.8 void clPalette::GenGreen3 ()

Generate black-green-yellow palette.

Definition at line 191 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clFrameBuf::PalGenGreen3().

7.69.3.9 void clPalette::GenGreen4 ()

Generate black-green-red-yellow palette.

Definition at line 210 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clFrameBuf::PalGenGreen4().

7.69.3.10 void clPalette::GenPureGreen ()

Generate black-green palette.

Definition at line 233 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clGUILocate::OnPaletteActivate(), and clFrameBuf::PalGenPureGreen().

7.69.3.11 void clPalette::GenWB ()

Generate black-white palette.

Definition at line 248 of file Palette.cc.

References iPalSize, Palette, and upPalette.

Referenced by clGUILocate::OnPaletteActivate(), and clFrameBuf::PalGenWB().

7.69.3.12 int clPalette::Size () `[inline]`

Get number of colors in palette.

Returns:
Number of colors

Definition at line 91 of file Palette.hh.

References iPalSize.

Referenced by clGUILocate::DisplayResults().

7.69.3.13 unsigned int clPalette::Color (int *iValue*) `[inline]`

Return color from palette LUT.

Parameters:
iValue Palette LUT index.

Returns:
Color

Definition at line 98 of file Palette.hh.

References upPalette.

7.69.3.14 unsigned int clPalette::operator[] (int *iValue*) `[inline]`

Return color from palette LUT.

Reimplemented in **clFrameBuf** (p. 228).

Definition at line 102 of file Palette.hh.

References upPalette.

7.69.4 Member Data Documentation

7.69.4.1 clAlloc clPalette::Palette `[private]`

Definition at line 35 of file Palette.hh.

Referenced by GenBW(), GenDir(), GenGreen(), GenGreen2(), GenGreen3(), GenGreen4(), GenHSV(), GenLight(), GenPureGreen(), GenTemp(), and GenWB().

7.69.4.2 int clPalette::iPalSize `[protected]`

Definition at line 37 of file Palette.hh.

Referenced by clPalette(), clFrameBuf::DrawColumn(), clFrameBuf::DrawLine(), GenBW(), GenDir(), GenGreen(), GenGreen2(), GenGreen3(), GenGreen4(), GenHSV(), GenLight(), GenPureGreen(), GenTemp(), GenWB(), clFrameBuf::GetNumColors(), and Size().

7.69.4.3 unsigned int* clPalette::upPalette `[protected]`

Definition at line 38 of file Palette.hh.

Referenced by clPalette(), Color(), clFrameBuf::DrawColumn(), clFrameBuf::DrawLine(), GenBW(), GenDir(), GenGreen(), GenGreen2(), GenGreen3(), GenGreen4(), GenHSV(), GenLight(), GenPureGreen(), GenTemp(), GenWB(), clFrameBuf::GetPalPtr(), and operator[]().

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

- **Palette.hh**
- **Palette.cc**

7.70 _stRawDataFirst Struct Reference

Header from streamdist server.

`#include <LocalMsg.h>`

Public Attributes

- int **iChannels**

  Number of channels.

- double **dSampleRate**

  Samplerate.

7.70.1 Detailed Description

Header from streamdist server.

Definition at line 29 of file LocalMsg.h.

7.70.2 Member Data Documentation

7.70.2.1 int _stRawDataFirst::iChannels

Number of channels.

Definition at line 31 of file LocalMsg.h.

Referenced by clStreamDist::AudioInThread(), clSaveSrv::ConnectStream(), clXMMSOut::CopyChannel(), clStreamDist::CopyChannel(), clSaveSrv::CreateFile(), clSaveSrv::CreateFile2(), clBeamSrv2Master::InDataThread(), clBeamSrvMaster::InDataThread(), clDirection::InitArray(), clBeamSrv2Master::InitProcessing(), clBeamSrvMaster::InitProcessing(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clXMMSOut::OpenAudio(), clDirection3::ProcessLoop(), clDirection2::ProcessLoop(), clBeamSrv2Master::ProcessLoop(), clBeamSrvMaster::ProcessLoop(), clBeamAudio::ProcessLoop(), clDirection::ReceiveData(), clXMMSOut::ServeClientThread(), clStreamDist::ServeClientThread(), clXMMSOut::WriteAudio(), and clSaveSrv::WriteData2().

7.70.2.2 double _stRawDataFirst::dSampleRate

Samplerate.

Definition at line 32 of file LocalMsg.h.

Referenced by clStreamDist::AudioInThread(), clSaveSrv::ConnectStream(), clSaveSrv::CreateFile(), clSaveSrv::CreateFile2(), clSpectrum::CreateFilter(), clDirection::InitArray(), clBeamAudio::InitBeam(), clDirection3::InitDir(), clDirection2::InitDir(), clLofarDemon::InitFilter(), clBeamSrv2Master::InitFilterBank(), clBeamSrvMaster::InitFilterBank(), clLevel::Initialize(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clLofarDemon::MainLoop(), clXMMSOut::OpenAudio(), clSpectrum::ProcessLoop(), clBeamAudio::SendFirst(), and clXMMSOut::WriteAudio().

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

- **LocalMsg.h**

7.71 _stRawDataReq Struct Reference

Request to streamdist server.

`#include <LocalMsg.h>`

Public Attributes

- int **iChannel**

  Channel or -1 for all.

7.71.1 Detailed Description

Request to streamdist server.

Definition at line 38 of file LocalMsg.h.

7.71.2 Member Data Documentation

7.71.2.1 int _stRawDataReq::iChannel

Channel or -1 for all.

Definition at line 40 of file LocalMsg.h.

Referenced by clSpectrum::ConnectStream(), clSaveSrv::ConnectStream(), clLevel::ConnectStream(), clDirection3::ConnectStream(), clDirection2::ConnectStream(), clBeamAudio::ConnectStream(), clLofarDemon::GetDataHdr(), clDirection::GetFirstMsg(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clXMMSOut::ServeClientThread(), and clStreamDist::ServeClientThread().

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

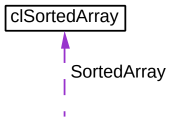
- **LocalMsg.h**

7.72 clRemoveNoise Class Reference

Background noise estimation and removal.

`#include <RemoveNoise.hh>`

Collaboration diagram for clRemoveNoise:



Public Member Functions

- **clRemoveNoise** ()
- ~**clRemoveNoise** ()
- void **TPSW** (float *, float, long, long, long)

  Two-Pass Split-Window algorithm.

- void **TPSW** (double *, double, long, long, long)
- void **TPSW** (float *, const float *, float, long, long, long)
- void **TPSW** (double *, const double *, double, long, long, long)
- void **OTA** (float *, float, long, long)

  Order-Truncate-Average algorithm.

- void **OTA** (double *, double, long, long)
- void **OTA** (float *, const float *, float, long, long)
- void **OTA** (double *, const double *, double, long, long)
- void **Diff** (float *, float, long)

  Differential method (experimental).

- void **Diff** (double *, double, long)
- void **Diff** (float *, const float *, float, long)
- void **Diff** (double *, const double *, double, long)
- void **InvDiff** (float *, float, long)

  Inverse differential method (experimental).

- void **InvDiff** (double *, double, long)
- void **InvDiff** (float *, const float *, float, long)
- void **InvDiff** (double *, const double *, double, long)

Private Member Functions

- void **GetPosSize** (long ∗, long ∗, long, long, long)
- void **GetPosSize1** (long ∗, long ∗, long, long, long, long)
- void **GetPosSize2** (long ∗, long ∗, long, long, long, long)
- long **GetAlphaMedian** (const float ∗, float, long)
- long **GetAlphaMedian** (const double ∗, double, long)

Private Attributes

- long **lPrevSize**
- clAlloc **PrevBuf**
- **clSortedArray SortedArray**

7.72.1 Detailed Description

Background noise estimation and removal.

Definition at line 105 of file RemoveNoise.hh.

7.72.2 Constructor & Destructor Documentation

7.72.2.1 clRemoveNoise::clRemoveNoise ()

Definition at line 218 of file RemoveNoise.cc.

References lPrevSize.

7.72.2.2 clRemoveNoise::~clRemoveNoise ()

Definition at line 224 of file RemoveNoise.cc.

7.72.3 Member Function Documentation

7.72.3.1 void clRemoveNoise::GetPosSize (long ∗, long ∗, long, long, long) `[inline, private]`

Definition at line 131 of file RemoveNoise.cc.

Referenced by OTA().

7.72.3.2 void clRemoveNoise::GetPosSize1 (long ∗, long ∗, long, long, long, long) `[inline, private]`

Definition at line 146 of file RemoveNoise.cc.

Referenced by TPSW().

7.72.3.3 void clRemoveNoise::GetPosSize2 (long *, long *, long, long, long, long) `[inline, private]`

Definition at line 163 of file RemoveNoise.cc.

Referenced by TPSW().

7.72.3.4 long clRemoveNoise::GetAlphaMedian (const float *, float, long) `[inline, private]`

Definition at line 180 of file RemoveNoise.cc.

Referenced by OTA().

7.72.3.5 long clRemoveNoise::GetAlphaMedian (const double *, double, long) `[inline, private]`

Definition at line 199 of file RemoveNoise.cc.

7.72.3.6 void clRemoveNoise::TPSW (float *, float, long, long, long)

Two-Pass Split-Window algorithm.

$$R_k = \{k - M, k - M + 1, ..., k - L, k + L, ..., k + M\}, 0 \leq L < M$$

$$K = \begin{cases} 2M + 1 & , L = 0 \\ 2M + 2 - 2L & , L \neq 0 \end{cases}$$

$$\hat{x}(k) = \frac{1}{K} \sum_{i \epsilon R_k} x(i)$$

$$y(k) = \begin{cases} x(k) & , x(k) \leq \alpha \hat{x}(k) \\ \hat{x}(k) & , x(k) > \alpha \hat{x}(k) \end{cases}$$

$$\hat{m}(k) = \frac{1}{K} \sum_{i \epsilon R_k} y(i)$$

then

$$z'(k) = \frac{x(k) - \hat{m}(k)}{\hat{m}(k)}, 0 \leq k \leq N - 1$$

Parameters:

fpVect Source&destination vector

fAlpha Alpha

lMeanLength Mean window length

lGapLength Gap length

lLength Length of source&destination vector

Definition at line 229 of file RemoveNoise.cc.

References GetPosSize1(), and GetPosSize2().

Referenced by clLofarDemon::CalcSpect(), clSpectDirLine2::Calculate(), clSpectDirLine::Calculate(), clSpectDirDipole2::Calculate(), clSpectDirDipole::Calculate(), clSpectrum::ProcessLoop(), and clDirection::SendResults().

7.72.3.7 void clRemoveNoise::TPSW (double ∗, double, long, long, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 272 of file RemoveNoise.cc.

References GetPosSize1(), and GetPosSize2().

7.72.3.8 void clRemoveNoise::TPSW (float ∗, const float ∗, float, long, long, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters:

fpDest Destination vector

fpSrc Source vector

lMeanLength Mean window length

lGapLength Gap length

lLength Length of source&destination vectors

Definition at line 315 of file RemoveNoise.cc.

References GetPosSize1(), and GetPosSize2().

7.72.3.9 void clRemoveNoise::TPSW (double ∗, const double ∗, double, long, long, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 358 of file RemoveNoise.cc.

References GetPosSize1(), and GetPosSize2().

7.72.3.10 void clRemoveNoise::OTA (float ∗, float, long, long)

Order-Truncate-Average algorithm.

Let y(k) be sorted (ascending) x(k) and Y_{sm} median, then

$$\hat{m}(k) = \frac{1}{I} \sum_{\substack{i=1 \\ i \epsilon R_k}}^{I} y(i), y(I) \leq \alpha Y_{sm}, y(I+1) > \alpha Y_{sm}$$

then

$$z'(k) = \frac{x(k) - \hat{m}(k)}{\hat{m}(k)}, 0 \leq k \leq N-1$$

Parameters:

fpVect Source&destination vector

fAlpha Alpha

lMeanLength Mean length

lLength Length of source&destination vector

Definition at line 401 of file RemoveNoise.cc.

References GetAlphaMedian(), and GetPosSize().

Referenced by clLofarDemon::CalcSpect(), clSpectDirLine2::Calculate(), clSpectDirLine::Calculate(), clSpectDirDipole2::Calculate(), clSpectDirDipole::Calculate(), clSpectrum::ProcessLoop(), and clDirection::SendResults().

7.72.3.11 void clRemoveNoise::OTA (double ∗, double, long, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 443 of file RemoveNoise.cc.

References GetAlphaMedian(), and GetPosSize().

7.72.3.12 void clRemoveNoise::OTA (float ∗, const float ∗, float, long, long)

Parameters:

fpDest Destination vector

fpSrc Source vector

fAlpha Alpha

lMeanLength Mean length

lLength Length of source&destination vectors

Definition at line 485 of file RemoveNoise.cc.

References GetAlphaMedian(), and GetPosSize().

7.72.3.13 void clRemoveNoise::OTA (double *, const double *, double, long, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 519 of file RemoveNoise.cc.

References GetAlphaMedian(), and GetPosSize().

7.72.3.14 void clRemoveNoise::Diff (float *, float, long)

Differential method (experimental).

$$z'(k) = x_1(k)w - x_2(k)(1-w), 0 \leq k \leq N-1$$

Parameters:

fpVect Source&destination vector

fWeight Weight, w

lLength Vector length

Definition at line 553 of file RemoveNoise.cc.

References lPrevSize, and PrevBuf.

Referenced by clLofarDemon::CalcSpect(), clSpectDirLine2::Calculate(), clSpectDirLine::Calculate(), clSpectDirDipole2::Calculate(), clSpectDirDipole::Calculate(), and clSpectrum::ProcessLoop().

7.72.3.15 void clRemoveNoise::Diff (double *, double, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 582 of file RemoveNoise.cc.

References lPrevSize, and PrevBuf.

7.72.3.16 void clRemoveNoise::Diff (float *, const float *, float, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters:

fpDest Destination vector

fpSrc Source vector

fWeight Weight, w

lLength Vector length

Definition at line 611 of file RemoveNoise.cc.

References lPrevSize, and PrevBuf.

7.72.3.17 void clRemoveNoise::Diff (double *, const double *, double, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 640 of file RemoveNoise.cc.

References lPrevSize, and PrevBuf.

7.72.3.18 void clRemoveNoise::InvDiff (float *, float, long)

Inverse differential method (experimental).

Definition at line 669 of file RemoveNoise.cc.

References lPrevSize, and PrevBuf.

Referenced by clLofarDemon::CalcSpect(), clSpectDirLine2::Calculate(), clSpectDirLine::Calculate(), clSpectDirDipole2::Calculate(), clSpectDirDipole::Calculate(), and clSpectrum::ProcessLoop().

7.72.3.19 void clRemoveNoise::InvDiff (double *, double, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 705 of file RemoveNoise.cc.

References lPrevSize, and PrevBuf.

7.72.3.20 void clRemoveNoise::InvDiff (float *, const float *, float, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 741 of file RemoveNoise.cc.

References lPrevSize, and PrevBuf.

7.72.3.21 void clRemoveNoise::InvDiff (double *, const double *, double, long)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 777 of file RemoveNoise.cc.

References lPrevSize, and PrevBuf.

7.72.4 Member Data Documentation

7.72.4.1 long clRemoveNoise::lPrevSize [private]

Definition at line 107 of file RemoveNoise.hh.

Referenced by clRemoveNoise(), Diff(), and InvDiff().

7.72.4.2 clAlloc clRemoveNoise::PrevBuf [private]

Definition at line 108 of file RemoveNoise.hh.

Referenced by Diff(), and InvDiff().

7.72.4.3 clSortedArray clRemoveNoise::SortedArray [private]

Definition at line 109 of file RemoveNoise.hh.

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

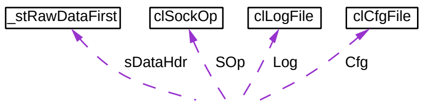
- **RemoveNoise.hh**
- **RemoveNoise.cc**

7.73 clSaveSrv Class Reference

Save server.

`#include <SaveSrv.hh>`

Collaboration diagram for clSaveSrv:



Public Member Functions

- **clSaveSrv** ()
- **~clSaveSrv** ()
- int **Main** (int, char **)
- void **Stop** ()

Public Attributes

- **clLogFile Log**

Private Member Functions

- bool **Initialize** ()
- bool **ConnectStream** ()
- bool **CreateFile** ()
- bool **CreateFile2** ()
- void **CreateFileName** (char *)
- void **ProcessLoop** ()
- void **Dither** ()
- bool **WriteData** ()
- bool **WriteData2** ()

Private Attributes

- bool **bRun**
- int **iFormat**
- int **iType**
- int **iBits**
- int **iDither**
- int **iFileTime**
- int **iFileH**

- unsigned int **uiRndSeed**
- long **lFrameLen**
- long **lFrameSize**
- char **cpSockName** [_POSIX_PATH_MAX+1]
- char **cpSavePath** [_POSIX_PATH_MAX+1]
- time_t **ttFileStarted**
- **stRawDataFirst sDataHdr**
- SNDFILE ∗ **spSndFile**
- clAlloc **InFrame**
- clAlloc **OutFrame**
- clAlloc **NoiseFrame**
- clAlloc **FLACFrame**
- **clCfgFile Cfg**
- clDSPOp **DSP**
- **clSockOp SOp**

7.73.1 Detailed Description

Save server.

Definition at line 79 of file SaveSrv.hh.

7.73.2 Constructor & Destructor Documentation

7.73.2.1 clSaveSrv::clSaveSrv ()

Definition at line 718 of file SaveSrv.cc.

References clLogFile::Add(), bRun, iFileH, Log, clLogFile::Open(), SAVS_-LOGFILE, and spSndFile.

7.73.2.2 clSaveSrv::∼clSaveSrv ()

Definition at line 734 of file SaveSrv.cc.

References clLogFile::Add(), iFileH, Log, and spSndFile.

7.73.3 Member Function Documentation

7.73.3.1 bool clSaveSrv::Initialize () `[private]`

Definition at line 126 of file SaveSrv.cc.

References clLogFile::Add(), Cfg, cpSavePath, cpSockName, FLACFrame, GDT, clCfgFile::GetInt(), clCfgFile::GetStr(), iBits, iDither, iFileTime, i-Format, InFrame, iType, lFrameLen, lFrameSize, Log, NoiseFrame, OutFrame, SAVS_CFGFILE, SAVS_DEF_FILETIME, SAVS_DEF_FRAMELEN, SAVS_-SNDFILE_MSGLEN, SD_CFGFILE, clCfgFile::SetFileName(), and uiRndSeed.

Referenced by Main().

7.73.3.2 bool clSaveSrv::ConnectStream () [private]

Definition at line 236 of file SaveSrv.cc.

References clLogFile::Add(), clSockClie::Connect(), cpSockName, _stRawDataFirst::dSampleRate, clSockOp::GetErrno(), clSockClie::GetErrno(), _stRawDataReq::iChannel, _stRawDataFirst::iChannels, Log, clSockOp::ReadN(), SAVS_LOGENTRY_SIZE, sDataHdr, clSockOp::SetHandle(), SOp, and clSockOp::WriteN().

Referenced by Main().

7.73.3.3 bool clSaveSrv::CreateFile () [private]

Definition at line 268 of file SaveSrv.cc.

References clLogFile::Add(), CreateFileName(), _stRawDataFirst::dSampleRate, iBits, _stRawDataFirst::iChannels, iDither, iFormat, iType, Log, SAVS_FORMAT_AIFF, SAVS_FORMAT_WAV, SAVS_TYPE_ADPCM, SAVS_TYPE_FLOAT, SAVS_TYPE_MSADPCM, SAVS_TYPE_PCM, sDataHdr, and spSndFile.

Referenced by ProcessLoop().

7.73.3.4 bool clSaveSrv::CreateFile2 () [private]

Definition at line 353 of file SaveSrv.cc.

References clLogFile::Add(), Cfg, CreateFileName(), _stRawDataFirst::dSampleRate, clCfgFile::GetInt(), iBits, _stRawDataFirst::iChannels, iFileH, lFrameLen, Log, and sDataHdr.

Referenced by ProcessLoop().

7.73.3.5 void clSaveSrv::CreateFileName (char *) [private]

Definition at line 456 of file SaveSrv.cc.

References cpFileExt, cpSavePath, iFileTime, iFormat, SAVS_TIMELEN, and ttFileStarted.

Referenced by CreateFile(), and CreateFile2().

7.73.3.6 void clSaveSrv::ProcessLoop () [private]

Definition at line 475 of file SaveSrv.cc.

References clLogFile::Add(), bRun, CreateFile(), CreateFile2(), clSockOp::GetErrno(), iFileTime, iFormat, InFrame, lFrameSize, Log, clSockOp::Read-

N(), SAVS_FORMAT_AIFF, SAVS_FORMAT_FLAC, SAVS_FORMAT_WAV, SAVS_SHUTDOWNFILE, SOp, Stop(), ttFileStarted, WriteData(), and WriteData2().

Referenced by Main().

7.73.3.7 void clSaveSrv::Dither () [private]

Definition at line 529 of file SaveSrv.cc.

References GDT, iBits, InFrame, lFrameLen, NoiseFrame, and OutFrame.

Referenced by WriteData(), and WriteData2().

7.73.3.8 bool clSaveSrv::WriteData () [private]

Definition at line 607 of file SaveSrv.cc.

References clLogFile::Add(), Dither(), GDT, iBits, iDither, InFrame, lFrameLen, Log, OutFrame, SAVS_SNDFILE_MSGLEN, and spSndFile.

Referenced by ProcessLoop().

7.73.3.9 bool clSaveSrv::WriteData2 () [private]

Definition at line 665 of file SaveSrv.cc.

References Dither(), DSP, FLACFrame, iBits, _stRawDataFirst::iChannels, iDither, InFrame, lFrameLen, OutFrame, and sDataHdr.

Referenced by ProcessLoop().

7.73.3.10 int clSaveSrv::Main (int, char **)

Definition at line 755 of file SaveSrv.cc.

References bDaemon, ConnectStream(), Initialize(), ProcessLoop(), SAVS_VERSMAJ, SAVS_VERSMIN, and SAVS_VERSPL.

Referenced by main().

7.73.3.11 void clSaveSrv::Stop () [inline]

Definition at line 123 of file SaveSrv.hh.

References bRun.

Referenced by ProcessLoop(), and SigHandler().

7.73.4 Member Data Documentation

7.73.4.1 bool clSaveSrv::bRun [private]

Definition at line 81 of file SaveSrv.hh.

Referenced by clSaveSrv(), ProcessLoop(), and Stop().

7.73.4.2 int clSaveSrv::iFormat [private]

Definition at line 82 of file SaveSrv.hh.

Referenced by CreateFile(), CreateFileName(), Initialize(), and ProcessLoop().

7.73.4.3 int clSaveSrv::iType [private]

Definition at line 83 of file SaveSrv.hh.

Referenced by CreateFile(), and Initialize().

7.73.4.4 int clSaveSrv::iBits [private]

Definition at line 84 of file SaveSrv.hh.

Referenced by CreateFile(), CreateFile2(), Dither(), Initialize(), WriteData(), and WriteData2().

7.73.4.5 int clSaveSrv::iDither [private]

Definition at line 85 of file SaveSrv.hh.

Referenced by CreateFile(), Initialize(), WriteData(), and WriteData2().

7.73.4.6 int clSaveSrv::iFileTime [private]

Definition at line 86 of file SaveSrv.hh.

Referenced by CreateFileName(), Initialize(), and ProcessLoop().

7.73.4.7 int clSaveSrv::iFileH [private]

Definition at line 87 of file SaveSrv.hh.

Referenced by clSaveSrv(), CreateFile2(), and ~clSaveSrv().

7.73.4.8 unsigned int clSaveSrv::uiRndSeed [private]

Definition at line 88 of file SaveSrv.hh.

Referenced by Initialize().

7.73.4.9 long clSaveSrv::lFrameLen [private]

Definition at line 89 of file SaveSrv.hh.

Referenced by CreateFile2(), Dither(), Initialize(), WriteData(), and WriteData2().

7.73.4.10 long clSaveSrv::lFrameSize [private]

Definition at line 90 of file SaveSrv.hh.

Referenced by Initialize(), and ProcessLoop().

7.73.4.11 char clSaveSrv::cpSockName[_POSIX_PATH_MAX + 1] [private]

Definition at line 91 of file SaveSrv.hh.

Referenced by ConnectStream(), and Initialize().

7.73.4.12 char clSaveSrv::cpSavePath[_POSIX_PATH_MAX + 1] [private]

Definition at line 92 of file SaveSrv.hh.

Referenced by CreateFileName(), and Initialize().

7.73.4.13 time_t clSaveSrv::ttFileStarted [private]

Definition at line 93 of file SaveSrv.hh.

Referenced by CreateFileName(), and ProcessLoop().

7.73.4.14 stRawDataFirst clSaveSrv::sDataHdr [private]

Definition at line 94 of file SaveSrv.hh.

Referenced by ConnectStream(), CreateFile(), CreateFile2(), and WriteData2().

7.73.4.15 SNDFILE* clSaveSrv::spSndFile [private]

Definition at line 95 of file SaveSrv.hh.

Referenced by clSaveSrv(), CreateFile(), WriteData(), and ~clSaveSrv().

7.73.4.16 clAlloc clSaveSrv::InFrame [private]

Definition at line 99 of file SaveSrv.hh.

Referenced by Dither(), Initialize(), ProcessLoop(), WriteData(), and WriteData2().

7.73.4.17 clAlloc clSaveSrv::OutFrame [private]

Definition at line 100 of file SaveSrv.hh.

Referenced by Dither(), Initialize(), WriteData(), and WriteData2().

7.73.4.18 clAlloc clSaveSrv::NoiseFrame [private]

Definition at line 101 of file SaveSrv.hh.

Referenced by Dither(), and Initialize().

7.73.4.19 clAlloc clSaveSrv::FLACFrame [private]

Definition at line 102 of file SaveSrv.hh.

Referenced by Initialize(), and WriteData2().

7.73.4.20 clCfgFile clSaveSrv::Cfg [private]

Definition at line 103 of file SaveSrv.hh.

Referenced by CreateFile2(), and Initialize().

7.73.4.21 clDSPOp clSaveSrv::DSP [private]

Definition at line 104 of file SaveSrv.hh.

Referenced by WriteData2().

7.73.4.22 clSockOp clSaveSrv::SOp [private]

Definition at line 105 of file SaveSrv.hh.

Referenced by ConnectStream(), and ProcessLoop().

7.73.4.23 clLogFile clSaveSrv::Log

Definition at line 119 of file SaveSrv.hh.

Referenced by clSaveSrv(), ConnectStream(), CreateFile(), CreateFile2(), Initialize(), ProcessLoop(), SigHandler(), WriteData(), and ~clSaveSrv().

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

- **SaveSrv.hh**
- **SaveSrv.cc**

7.74 _stSGramInfo Struct Reference

Information stored in saved spectrogram .tif.inf.

`#include <ConvPic.hh>`

Public Attributes

- time_t **ttTime**

  Newest dataline time.

- double **dLowFreq**

  Lower frequency limit.

- double **dHighFreq**

  Upper frequency limit.

- double **dLineTime**

  Length of one column in time (s).

7.74.1 Detailed Description

Information stored in saved spectrogram .tif.inf.

Definition at line 85 of file ConvPic.hh.

7.74.2 Member Data Documentation

7.74.2.1 time_t _stSGramInfo::ttTime

Newest dataline time.

Definition at line 87 of file ConvPic.hh.

Referenced by clConvPic::DrawSGramInfo(), and clConvPic::LoadSGramInfo().

7.74.2.2 double _stSGramInfo::dLowFreq

Lower frequency limit.

Definition at line 88 of file ConvPic.hh.

Referenced by clConvPic::DrawSGramInfo(), and clConvPic::LoadSGramInfo().

7.74.2.3 double _stSGramInfo::dHighFreq

Upper frequency limit.

Definition at line 89 of file ConvPic.hh.

Referenced by clConvPic::DrawSGramInfo(), and clConvPic::LoadSGramInfo().

7.74.2.4 double _stSGramInfo::dLineTime

Length of one column in time (s).

Definition at line 90 of file ConvPic.hh.

Referenced by clConvPic::DrawSGramInfo(), and clConvPic::LoadSGramInfo().

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

- **ConvPic.hh**

7.75 clSockClie Class Reference

Socket client class.

`#include <SockClie.hh>`

Public Member Functions

- **clSockClie** ()
- **∼clSockClie** ()
- int **Connect** (const char ∗, const char ∗, int)

  Connect to TCP server, set either hostname or hostaddr and the other to NULL.

- int **Connect** (const char ∗)

  Connect to Unix domain server.

- int **GetErrno** ()

  Get errno.

Private Attributes

- int **iErrno**

7.75.1 Detailed Description

Socket client class.

Definition at line 34 of file SockClie.hh.

7.75.2 Constructor & Destructor Documentation

7.75.2.1 clSockClie::clSockClie ()

Definition at line 43 of file SockClie.cc.

References iErrno.

7.75.2.2 clSockClie::∼clSockClie ()

Definition at line 49 of file SockClie.cc.

7.75.3 Member Function Documentation

7.75.3.1 int clSockClie::Connect (const char *, const char *, int)

Connect to TCP server, set either hostname or hostaddr and the other to NULL.

Parameters:

cpHostName Hostname

cpHostAddr Host address

iHostPort Host port

Returns:

Handle, -1 on error

Definition at line 54 of file SockClie.cc.

References iErrno.

Referenced by clStreamDist::AudioInThread(), Connect(), clSubLocate::ConnectDir(), clSpectrum::ConnectStream(), clSaveSrv::ConnectStream(), clLevel::ConnectStream(), clDirection3::ConnectStream(), clDirection2::ConnectStream(), clBeamAudio::ConnectStream(), clGUILocate::ConnectToServer(), clGUILevel::ConnectToServer(), clDirection::Exec(), clLofarDemon::GetDataHdr(), clSpectGUI::InitConnection(), clGUILofar::InitConnection(), clGUIDir::InitConnection(), clBeamAudioUI::InitConnection(), main(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clSoundUI::OnClickedEvent(), and clSoundProxy::SoundInThread().

7.75.3.2 int clSockClie::Connect (const char *)

Connect to Unix domain server.

In QNX (which doesn't support domain sockets) this is faked to TCP socket and filename should be port number.

Parameters:

cpSockName Socket filename

Returns:

Handle, -1 on error

Definition at line 114 of file SockClie.cc.

References Connect(), GLOBAL_SOCKET_PATH, iErrno, and UNIX_PATH_MAX.

7.75.3.3 int clSockClie::GetErrno () `[inline]`

Get errno.

Returns:
errno

Definition at line 65 of file SockClie.hh.

References iErrno.

Referenced by clStreamDist::AudioInThread(), clSaveSrv::ConnectStream(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), and clSoundProxy::SoundInThread().

7.75.4 Member Data Documentation

7.75.4.1 int clSockClie::iErrno `[private]`

Definition at line 36 of file SockClie.hh.

Referenced by clSockClie(), Connect(), and GetErrno().

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

- **SockClie.hh**
- **SockClie.cc**

7.76 clSockOp Class Reference

Socket I/O operations.

`#include <SockOp.hh>`

Public Member Functions

- **clSockOp** ()
- **clSockOp** (int)
- ~**clSockOp** ()
- void **SetHandle** (int)

  Set handle.

- void **Close** ()

  Close handle/connection.

- bool **ReadSelect** (int)

  Wait for socket to become readable.

- bool **WriteSelect** (int)

  Wait for socket to become writable.

- int **Read** (void *, int)

  Read data.

- int **Write** (const void *, int)

  Write data.

- int **ReadN** (void *, int)

  Read N bytes.

- int **WriteN** (const void *, int)

  Write N bytes.

- int **Receive** (void *, int, int)

  Receive data.

- int **Send** (const void *, int, int)

  Send data.

- int **ReceiveMsg** (struct msghdr *, int)

  Receive message.

- int **SendMsg** (const struct msghdr *, int)

  Send message.

- int **Shutdown** (int)
  Shutdown connection.
- int **GetSockName** (struct sockaddr *, socklen_t *)
  Get address of this end.
- int **GetPeerName** (struct sockaddr *, socklen_t *)
  Get address of remote end.
- bool **SetBlocking** (bool)
  Set blocking/nonblocking IO.
- bool **SetKeepAlive** ()
  Enable keepalive messages.
- bool **SetLinger** (int)
  Enable linger option.
- int **GetRecvBufSize** ()
  Get receive buffer size.
- bool **SetRecvBufSize** (int)
  Set receive buffer size.
- int **GetSendBufSize** ()
  Get send buffer size.
- bool **SetSendBufSize** (int)
  Set send buffer size.
- int **GetRecvBufLowWater** ()
  Get position of receive buffer low-water sign.
- bool **SetRecvBufLowWater** (int)
  Set receive buffer low-water sign.
- int **GetSendBufLowWater** ()
  Get position of send buffer low-water sign.
- bool **SetSendBufLowWater** (int)
  Set send buffer low-water sign.
- bool **SetRecvTimeout** (int)
  Set receive timeout.

- bool **SetSendTimeout** (int)

  Set send timeout.

- bool **SetTCPKeepAlive** (int)

  Set interval of TCP keepalive messages.

- bool **DisableNagle** ()

  Disable TCP nagle-algorithm.

- bool **SetNagle** (bool)
- bool **SetTypeOfService** (int)

  Set type-of-service flag.

- void **SetCloseOnDestruct** (bool)

  Close handle/connection in destructor.

- int **GetHandle** ()

  Get handle (file descriptor).

- int **GetErrno** ()

  Get errno.

Private Attributes

- int **iErrno**
- int **iRetVal**
- int **iSockH**
- bool **bCloseOnDestruct**

7.76.1 Detailed Description

Socket I/O operations.

Definition at line 39 of file SockOp.hh.

7.76.2 Constructor & Destructor Documentation

7.76.2.1 clSockOp::clSockOp ()

Definition at line 38 of file SockOp.cc.

References bCloseOnDestruct, and iSockH.

7.76.2.2 clSockOp::clSockOp (int)

Definition at line 45 of file SockOp.cc.

References bCloseOnDestruct, and iSockH.

7.76.2.3 clSockOp::~clSockOp ()

Definition at line 52 of file SockOp.cc.

References bCloseOnDestruct, and iSockH.

7.76.3 Member Function Documentation

7.76.3.1 void clSockOp::SetHandle (int)

Set handle.

Closes previously open handle.

Parameters:
iHandle Handle

Definition at line 58 of file SockOp.cc.

References Close(), and iSockH.

Referenced by clStreamDist::AudioInThread(), clBeamAudio::clBeamAudio(), clDirection2::clDirection2(), clDirection3::clDirection3(), clLevel::clLevel(), clLofarDemon::clLofarDemon(), clSpectrum::clSpectrum(), clSubLocate::ConnectDir(), clSpectrum::ConnectStream(), clSaveSrv::ConnectStream(), clLevel::ConnectStream(), clDirection3::ConnectStream(), clDirection2::ConnectStream(), clBeamAudio::ConnectStream(), clGUILocate::ConnectToServer(), clGUILevel::ConnectToServer(), clLofarDemon::GetDataHdr(), clGUILofar::InitConnection(), clBeamAudioUI::InitConnection(), main(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clSoundProxy::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), clSoundUI::SoundInThread(), clSoundProxy::SoundInThread(), and clUIServer::Wait().

7.76.3.2 void clSockOp::Close ()

Close handle/connection.

Definition at line 65 of file SockOp.cc.

References iSockH.

Referenced by clGUILofar::OnConnectClick(), clGUILocate::OnConnectClick(), clGUILevel::OnConnectClick(), clBeamAudioUI::OnConnectClick(), clGUILofar::OnGdkInput(), clGUILocate::OnGdkInput(), clGUILevel::OnGdk-Input(), clSoundProxy::ServeClientThread(), SetHandle(), clSoundUI::Sound-InThread(), and clSoundProxy::SoundInThread().

7.76.3.3 bool clSockOp::ReadSelect (int)

Wait for socket to become readable.

Parameters:
  iTimeoutMS Timeout (ms)

Returns:
  Readable?

Definition at line 75 of file SockOp.cc.

References iErrno, and iSockH.

Referenced by clSpectrum::ConnectStream(), clLevel::ConnectStream(), clDirection3::ConnectStream(), clDirection2::ConnectStream(), clBeam-Audio::ConnectStream(), clSpectGUI::GdkInput(), clLofarDemon::GetData-Hdr(), clDirection::GetFirstMsg(), clLevel::GetRequest(), clDirection::Get-Request(), clLofarDemon::GetRequestMsg(), clSpectrum::GetRq(), cl-Direction3::GetRq(), clDirection2::GetRq(), clBeamAudio::GetRq(), main(), clGUILofar::OnGdkInput(), clGUILocate::OnGdkInput(), clGUILevel::On-GdkInput(), clBeamAudioUI::OnGdkInput(), clBeamAudio::ProcessLoop(), clLevel::ReceiveData(), clDirection::ReceiveData(), clBeamAudioUI::Send-Settings(), clSoundUI::SoundInThread(), and clSoundProxy::SoundInThread().

7.76.3.4 bool clSockOp::WriteSelect (int)

Wait for socket to become writable.

Parameters:
  iTimeoutMS Timeout (ms)

Returns:
  Writable?

Definition at line 100 of file SockOp.cc.

References iErrno, and iSockH.

Referenced by clLofarDemon::SendResults(), clDirection::SendResults(), cl-SoundProxy::ServeClientThread(), and clBeamAudioUI::UpdateSettings().

7.76.3.5 int clSockOp::Read (void *, int)

Read data.

Parameters:

vpBuf Buffer

iBufLen Buffer size

Returns:

Bytes read

Definition at line 125 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

Referenced by main().

7.76.3.6 int clSockOp::Write (const void *, int)

Write data.

Parameters:

vpBuf Buffer

iBufLen Buffer size

Returns:

Bytes written

Definition at line 133 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

7.76.3.7 int clSockOp::ReadN (void *, int)

Read N bytes.

Ensures that N bytes are read, unless error occurs.

Parameters:

vpBuf Buffer

iBufLen Buffer size

Returns:

Bytes read

Definition at line 141 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

Referenced by clStreamDist::AudioInThread(), clSpectrum::ConnectStream(), clSaveSrv::ConnectStream(), clLevel::ConnectStream(), clDirection3::ConnectStream(), clDirection2::ConnectStream(), clBeamAudio::ConnectStream(), clGUILocate::ConnectToServer(), clSpectGUI::GdkInput(), clSpectrum::GetData(), clLofarDemon::GetDataHdr(), clDirection::GetFirstMsg(), clLevel::GetRequest(), clDirection::GetRequest(), clLofarDemon::GetRequestMsg(), clSpectrum::GetRq(), clDirection3::GetRq(), clDirection2::GetRq(), clBeamAudio::GetRq(), clBeamSrv2Master::InDataThread(), clBeamSrvMaster::InDataThread(), main(), clSubLocate::Main(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clLofarDemon::MainLoop(), clGUILofar::OnGdkInput(), clGUILocate::OnGdkInput(), clGUILevel::OnGdkInput(), clGUIDir::OnGdkInput(), clBeamAudioUI::OnGdkInput(), clSaveSrv::ProcessLoop(), clDirection3::ProcessLoop(), clDirection2::ProcessLoop(), clBeamAudio::ProcessLoop(), clLevel::ReceiveData(), clDirection::ReceiveData(), clBeamAudioUI::SendSettings(), clXMMSOut::ServeClientThread(), clStreamDist::ServeClientThread(), clSoundUI::SoundInThread(), and clSoundProxy::SoundInThread().

7.76.3.8 int clSockOp::WriteN (const void *, int)

Write N bytes.

Ensures that N bytes are written, unless error occurs.

Parameters:

vpBuf Buffer

iBufLen Buffer size

Returns:

Bytes written

Definition at line 160 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

Referenced by clSubLocate::ConnectDir(), clSpectrum::ConnectStream(), clSaveSrv::ConnectStream(), clLevel::ConnectStream(), clDirection3::ConnectStream(), clDirection2::ConnectStream(), clBeamAudio::ConnectStream(), clGUILevel::ConnectToServer(), clLofarDemon::GetDataHdr(), clDirection::GetFirstMsg(), clSpectGUI::InitConnection(), clGUILofar::InitConnection(), clGUIDir::InitConnection(), clBeamAudioUI::InitConnection(), main(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clSpectrum::ProcessLoop(), clDirection3::ProcessLoop(), clDirection2::ProcessLoop(), clBeamAudio::SendFirst(), clLevel::SendResult(), clBeamAudio::SendResult(), clLofarDemon::SendResults(), clDirection::SendResults(), clSpectGUI::SendSettings(), clGUILofar::SendSettings(), clGUILevel::SendSettings(), clGUIDir::SendSettings(), clBeamAudioUI::SendSettings(), clXMMSOut::ServeClientThread(), clStreamDist::ServeClientThread(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clSoundProxy::ServeClientThread(), clLocate::ServeClientThread(), clFileSrv::ServeClientThread(), clComediSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), and clBeamAudioUI::UpdateSettings().

7.76.3.9 int clSockOp::Receive (void $*$, int, int)

Receive data.

Parameters:

vpBuf Buffer

iBufLen Buffer size

iFlags Flags

Returns:

Bytes received

Definition at line 180 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

7.76.3.10 int clSockOp::Send (const void $*$, int, int)

Send data.

Parameters:

vpBuf Buffer

iBufLen Buffer size

iFlags Flags

Returns:

Bytes sent

Definition at line 188 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

7.76.3.11 int clSockOp::ReceiveMsg (struct msghdr $*$, int)

Receive message.

Parameters:

spMsgHdr Message header

iFlags Flags

Returns:

Bytes received

Definition at line 196 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

7.76.3.12 int clSockOp::SendMsg (const struct msghdr *, int)

Send message.

Parameters:
- ***spMsgHdr*** Message header
- ***iFlags*** Flags

Returns:
Bytes sent

Definition at line 204 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

7.76.3.13 int clSockOp::Shutdown (int)

Shutdown connection.

Parameters:
- ***iShutdownDir*** Direction flag

Returns:
0 on success, -1 on error

Definition at line 212 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

Referenced by main(), clGUILofar::OnConnectClick(), clGUIDir::OnConnectClick(), clGUILofar::OnGdkInput(), clGUIDir::OnGdkInput(), and clSoundProxy::SoundInThread().

7.76.3.14 int clSockOp::GetSockName (struct sockaddr *, socklen_t *)

Get address of this end.

Parameters:
- ***spAddr*** Address
- ***ipAddrLen*** Address length

Returns:
0 on success, -1 on error

Definition at line 220 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

7.76.3.15 int clSockOp::GetPeerName (struct sockaddr *, socklen_t *)

Get address of remote end.

Parameters:

spAddr Address

ipAddrLen Address length

Returns:

0 on succes, -1 on error

Definition at line 228 of file SockOp.cc.

References iErrno, iRetVal, and iSockH.

Referenced by clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clSoundProxy::ServeClientThread(), clComediSrv::ServeClientThread(), and clUIServer::Wait().

7.76.3.16 bool clSockOp::SetBlocking (bool)

Set blocking/nonblocking IO.

Parameters:

bBlocking Blocking?

Returns:

Success

Definition at line 236 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.17 bool clSockOp::SetKeepAlive ()

Enable keepalive messages.

Returns:

Success

Definition at line 263 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.18 bool clSockOp::SetLinger (int)

Enable linger option.

Parameters:
iLingerTime Linger time

Returns:
Success

Definition at line 277 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.19 int clSockOp::GetRecvBufSize ()

Get receive buffer size.

Returns:
Size of receive buffer

Definition at line 293 of file SockOp.cc.

References iErrno, and iSockH.

Referenced by clLofarDemon::MainLoop().

7.76.3.20 bool clSockOp::SetRecvBufSize (int)

Set receive buffer size.

Parameters:
iBufSize Size of receive buffer

Returns:
Success

Definition at line 309 of file SockOp.cc.

References iErrno, and iSockH.

Referenced by clSubLocate::Main(), clLofarDemon::MainLoop(), clDirection3::ProcessLoop(), clDirection2::ProcessLoop(), and clSoundUI::SoundInThread().

7.76.3.21 int clSockOp::GetSendBufSize ()

Get send buffer size.

Returns:
Size of send buffer

Definition at line 321 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.22 bool clSockOp::SetSendBufSize (int)

Set send buffer size.

Parameters:
iBufSize Size of send buffer

Returns:
Success

Definition at line 337 of file SockOp.cc.

References iErrno, and iSockH.

Referenced by clDirection3::ProcessLoop(), clDirection2::ProcessLoop(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), and clComediSrv::ServeClientThread().

7.76.3.23 int clSockOp::GetRecvBufLowWater ()

Get position of receive buffer low-water sign.

Returns:
Low-water position

Definition at line 349 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.24 bool clSockOp::SetRecvBufLowWater (int)

Set receive buffer low-water sign.

Parameters:
iBufLowWater Position of receive low-water sign

Returns:
Success

Definition at line 365 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.25 int clSockOp::GetSendBufLowWater ()

Get position of send buffer low-water sign.

Returns:
Low-water position

Definition at line 377 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.26 bool clSockOp::SetSendBufLowWater (int)

Set send buffer low-water sign.

Parameters:
: ***iBufLowWater*** Position of send low-water sign

Returns:
: Success

Definition at line 393 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.27 bool clSockOp::SetRecvTimeout (int)

Set receive timeout.

Parameters:
: ***iTimeoutMS*** Timeout (ms)

Returns:
: Success

Definition at line 405 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.28 bool clSockOp::SetSendTimeout (int)

Set send timeout.

Parameters:
: ***iTimeoutMS*** Timeout (ms)

Returns:
: Success

Definition at line 421 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.29 bool clSockOp::SetTCPKeepAlive (int)

Set interval of TCP keepalive messages.

Parameters:
: ***iTimeSec*** Time between keepalive messages (s)

Returns:
Success

Definition at line 437 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.30 bool clSockOp::DisableNagle ()

Disable TCP nagle-algorithm.

Returns:
Success

Definition at line 451 of file SockOp.cc.

References iErrno, and iSockH.

Referenced by clBeamAudio::clBeamAudio(), main(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clSoundProxy::ServeClientThread(), clFileSrv::ServeClientThread(), clComediSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), and clSoundProxy::SoundInThread().

7.76.3.31 bool clSockOp::SetNagle (bool)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 465 of file SockOp.cc.

References iErrno, and iSockH.

7.76.3.32 bool clSockOp::SetTypeOfService (int)

Set type-of-service flag.

Parameters:
iTOS Type-of-service flag (IPTOS_*)

Returns:
Success

Definition at line 479 of file SockOp.cc.

References iErrno, and iSockH.

Referenced by main(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clSoundProxy::ServeClientThread(), clFileSrv::ServeClientThread(), clComediSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), and clSoundProxy::SoundInThread().

7.76.3.33 void clSockOp::SetCloseOnDestruct (bool)

Close handle/connection in destructor.

Parameters:
bSetting Close in destructor?

Definition at line 491 of file SockOp.cc.

References bCloseOnDestruct.

Referenced by clUIServer::Wait().

7.76.3.34 int clSockOp::GetHandle () `[inline]`

Get handle (file descriptor).

Returns:
Handle

Definition at line 287 of file SockOp.hh.

References iSockH.

Referenced by clGUILocate::OnConnectClick(), and clGUILevel::OnConnect-Click().

7.76.3.35 int clSockOp::GetErrno () `[inline]`

Get errno.

Returns:
errno

Definition at line 293 of file SockOp.hh.

References iErrno.

Referenced by clStreamDist::AudioInThread(), clSaveSrv::ConnectStream(), cl-SpectGUI::GdkInput(), clDirection::GetFirstMsg(), clDirection::GetRequest(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clGUILofar::On-GdkInput(), clGUIDir::OnGdkInput(), clBeamAudioUI::OnGdkInput(), clSaveSrv::ProcessLoop(), clDirection::ReceiveData(), clLofarDemon::Send-Results(), clDirection::SendResults(), clGUIDir::SendSettings(), clSound-Proxy::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), cl-BeamSrvMaster::ServeClientThread(), clSoundUI::SoundInThread(), clSound-Proxy::SoundInThread(), and clBeamAudioUI::UpdateSettings().

7.76.4 Member Data Documentation

7.76.4.1 int clSockOp::iErrno [private]

Definition at line 41 of file SockOp.hh.

Referenced by DisableNagle(), GetErrno(), GetPeerName(), GetRecvBufLowWater(), GetRecvBufSize(), GetSendBufLowWater(), GetSendBufSize(), GetSockName(), Read(), ReadN(), ReadSelect(), Receive(), ReceiveMsg(), Send(), SendMsg(), SetBlocking(), SetKeepAlive(), SetLinger(), SetNagle(), SetRecvBufLowWater(), SetRecvBufSize(), SetRecvTimeout(), SetSendBufLowWater(), SetSendBufSize(), SetSendTimeout(), SetTCPKeepAlive(), SetTypeOfService(), Shutdown(), Write(), WriteN(), and WriteSelect().

7.76.4.2 int clSockOp::iRetVal [private]

Definition at line 42 of file SockOp.hh.

Referenced by GetPeerName(), GetSockName(), Read(), ReadN(), Receive(), ReceiveMsg(), Send(), SendMsg(), Shutdown(), Write(), and WriteN().

7.76.4.3 int clSockOp::iSockH [private]

Definition at line 43 of file SockOp.hh.

Referenced by Close(), clSockOp(), DisableNagle(), GetHandle(), GetPeerName(), GetRecvBufLowWater(), GetRecvBufSize(), GetSendBufLowWater(), GetSendBufSize(), GetSockName(), Read(), ReadN(), ReadSelect(), Receive(), ReceiveMsg(), Send(), SendMsg(), SetBlocking(), SetHandle(), SetKeepAlive(), SetLinger(), SetNagle(), SetRecvBufLowWater(), SetRecvBufSize(), SetRecvTimeout(), SetSendBufLowWater(), SetSendBufSize(), SetSendTimeout(), SetTCPKeepAlive(), SetTypeOfService(), Shutdown(), Write(), WriteN(), WriteSelect(), and ~clSockOp().

7.76.4.4 bool clSockOp::bCloseOnDestruct [private]

Definition at line 44 of file SockOp.hh.

Referenced by clSockOp(), SetCloseOnDestruct(), and ~clSockOp().

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

- **SockOp.hh**
- **SockOp.cc**

7.77 clSockServ Class Reference

Socket server class.

`#include <SockServ.hh>`

Public Member Functions

- **clSockServ** ()
- **clSockServ** (unsigned short)
- **clSockServ** (const char ∗)
- ∼**clSockServ** ()
- bool **Bind** (unsigned short)

  Bind to TCP port.
- bool **Bind** (const char ∗, unsigned short)

  Bind to specific address and port.
- bool **Bind** (const char ∗)

  Bind to UNIX domain socket.
- int **WaitForConnect** ()

  Wait for connect.
- int **WaitForConnect** (int)

  Wait for connect.
- void **Close** ()

  Close listening socket.
- int **GetErrno** ()

  Get errno.

Private Attributes

- int **iErrno**
- int **iListenH**
- bool **bLocal**
- char **cpLocalSockName** [UNIX_PATH_MAX+1]

7.77.1 Detailed Description

Socket server class.

Definition at line 44 of file SockServ.hh.

7.77.2 Constructor & Destructor Documentation

7.77.2.1 clSockServ::clSockServ ()

Definition at line 41 of file SockServ.cc.

References iErrno, and iListenH.

7.77.2.2 clSockServ::clSockServ (unsigned *short*)

Definition at line 48 of file SockServ.cc.

References Bind(), and iErrno.

7.77.2.3 clSockServ::clSockServ (const char ∗)

Definition at line 55 of file SockServ.cc.

References Bind(), and iErrno.

7.77.2.4 clSockServ::∼clSockServ ()

Definition at line 195 of file SockServ.cc.

References Close().

7.77.3 Member Function Documentation

7.77.3.1 bool clSockServ::Bind (unsigned *short*)

Bind to TCP port.

Parameters:
usPort Port number

Returns:
Success

Definition at line 63 of file SockServ.cc.

References bLocal, iErrno, iListenH, and SOCKSERV_LISTENQUEUE_LEN.

Referenced by Bind(), clSockServ(), clXMMSOut::Init(), clStreamDist::Main(), clSoundSrvA::Main(), clSoundSrv2::Main(), clLocate::Main(), clComedi-Srv::Main(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), clFile-Srv::ServerThread(), and clSoundProxy::WaitConnectThread().

7.77.3.2 bool clSockServ::Bind (const char ∗, unsigned *short*)

Bind to specific address and port.

Parameters:

cpAddress Address

usPort Port

Returns:

Success

Definition at line 99 of file SockServ.cc.

References bLocal, iErrno, iListenH, and SOCKSERV_LISTENQUEUE_LEN.

7.77.3.3 bool clSockServ::Bind (const char *)

Bind to UNIX domain socket.

Parameters:

cpSockFile Filename

Returns:

Success

Definition at line 142 of file SockServ.cc.

References Bind(), bLocal, cpLocalSockName, GLOBAL_SOCKET_PATH, iErrno, iListenH, SOCKSERV_LISTENQUEUE_LEN, and UNIX_PATH_MAX.

7.77.3.4 int clSockServ::WaitForConnect ()

Wait for connect.

Note:

This has no timeout.

Returns:

Handle to new connection, -1 on error

Definition at line 201 of file SockServ.cc.

References bLocal, iErrno, and iListenH.

Referenced by clStreamDist::Main(), clSoundSrvA::Main(), clSoundSrv2::Main(), clLocate::Main(), clComediSrv::Main(), clXMMSOut::MainThread(), clFileSrv::ServerThread(), clBeamSrv2Master::ServerThread(), clBeamSrvMaster::ServerThread(), clUIServer::Wait(), and clSoundProxy::WaitConnectThread().

7.77.3.5 int clSockServ::WaitForConnect (int)

Wait for connect.

Note:
Check errno for timeout/error.

Parameters:
iTimeout Timeout (ms)

Returns:
Handle to new connection, -1 on error

Definition at line 229 of file SockServ.cc.

References bLocal, iErrno, and iListenH.

7.77.3.6 void clSockServ::Close ()

Close listening socket.

Definition at line 278 of file SockServ.cc.

References bLocal, cpLocalSockName, and iListenH.

Referenced by clXMMSOut::OnButtonClick(), and ~clSockServ().

7.77.3.7 int clSockServ::GetErrno () `[inline]`

Get errno.

Returns:
errno

Definition at line 103 of file SockServ.hh.

References iErrno.

Referenced by clUIServer::clUIServer(), clStreamDist::Main(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), and clUIServer::Wait().

7.77.4 Member Data Documentation

7.77.4.1 int clSockServ::iErrno `[private]`

Definition at line 46 of file SockServ.hh.

Referenced by Bind(), clSockServ(), GetErrno(), and WaitForConnect().

7.77.4.2 int clSockServ::iListenH [private]

Definition at line 47 of file SockServ.hh.

Referenced by Bind(), Close(), clSockServ(), and WaitForConnect().

7.77.4.3 bool clSockServ::bLocal [private]

Definition at line 48 of file SockServ.hh.

Referenced by Bind(), Close(), and WaitForConnect().

7.77.4.4 char clSockServ::cpLocalSockName[UNIX_PATH_MAX + 1] [private]

Definition at line 49 of file SockServ.hh.

Referenced by Bind(), and Close().

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

- **SockServ.hh**
- **SockServ.cc**

7.78 clSortedArray Class Reference

Class for array of sorted values with insert/remove functionality.

`#include <RemoveNoise.hh>`

Public Member Functions

- **clSortedArray** ()
- ~**clSortedArray** ()
- void **Initialize** (const float *, long)

  Initialize array, optionally with data.

- void **InsertTail** (float)

  Insert new value to "tail" of the array.

- void **RemoveHead** ()

  Remove value from "head" of the array.

- void **GetValues** (float *, long)

  Get values in array (in sorted order).

- long **GetSize** ()

  Get number of values in array.

- void **Clear** ()

  Clear the array.

Private Types

- enum **eDataType** { **TYPE_DOUBLE**, **TYPE_FLOAT** }

Private Attributes

- **eDataType DataType**
- long **lFrontSerial**
- long **lTailSerial**
- **ValueMap_t mapValues**
- **ValueMapf_t mapValuesf**

7.78.1 Detailed Description

Class for array of sorted values with insert/remove functionality.

Definition at line 40 of file RemoveNoise.hh.

7.78.2 Member Enumeration Documentation

7.78.2.1 enum clSortedArray::eDataType [private]

Enumeration values:

TYPE_DOUBLE

TYPE_FLOAT

Definition at line 42 of file RemoveNoise.hh.

7.78.3 Constructor & Destructor Documentation

7.78.3.1 clSortedArray::clSortedArray ()

Definition at line 35 of file RemoveNoise.cc.

References lFrontSerial, and lTailSerial.

7.78.3.2 clSortedArray::~clSortedArray ()

Definition at line 42 of file RemoveNoise.cc.

7.78.4 Member Function Documentation

7.78.4.1 void clSortedArray::Initialize (const float *, long)

Initialize array, optionally with data.

Parameters:

fpValues Initialization values

lValueCount Number of values in initialization array

Definition at line 47 of file RemoveNoise.cc.

References Clear(), DataType, InsertTail(), and TYPE_FLOAT.

7.78.4.2 void clSortedArray::InsertTail (float)

Insert new value to "tail" of the array.

Parameters:

fValue Value to insert

Definition at line 59 of file RemoveNoise.cc.

References lTailSerial, and mapValuesf.

Referenced by Initialize().

7.78.4.3 void clSortedArray::RemoveHead ()

Remove value from "head" of the array.

Definition at line 66 of file RemoveNoise.cc.

References DataType, lFrontSerial, mapValuesf, and TYPE_FLOAT.

7.78.4.4 void clSortedArray::GetValues (float $*$, long)

Get values in array (in sorted order).

Parameters:

fpValues Destination array

lValueCount Size of destination array

Exceptions:

clXSortedArray

Definition at line 87 of file RemoveNoise.cc.

References DataType, GetSize(), mapValuesf, and TYPE_FLOAT.

7.78.4.5 long clSortedArray::GetSize ()

Get number of values in array.

Returns:

Number of values

Exceptions:

clXSortedArray

Definition at line 109 of file RemoveNoise.cc.

References DataType, mapValues, mapValuesf, TYPE_DOUBLE, and TYPE_-FLOAT.

Referenced by GetValues().

7.78.4.6 void clSortedArray::Clear ()

Clear the array.

Definition at line 122 of file RemoveNoise.cc.

References lFrontSerial, lTailSerial, mapValues, and mapValuesf.

Referenced by Initialize().

7.78.5 Member Data Documentation

7.78.5.1 eDataType clSortedArray::DataType [private]

Definition at line 55 of file RemoveNoise.hh.

Referenced by GetSize(), GetValues(), Initialize(), and RemoveHead().

7.78.5.2 long clSortedArray::lFrontSerial [private]

Definition at line 56 of file RemoveNoise.hh.

Referenced by Clear(), clSortedArray(), and RemoveHead().

7.78.5.3 long clSortedArray::lTailSerial [private]

Definition at line 57 of file RemoveNoise.hh.

Referenced by Clear(), clSortedArray(), and InsertTail().

7.78.5.4 ValueMap_t clSortedArray::mapValues [private]

Definition at line 58 of file RemoveNoise.hh.

Referenced by Clear(), and GetSize().

7.78.5.5 ValueMapf_t clSortedArray::mapValuesf [private]

Definition at line 59 of file RemoveNoise.hh.

Referenced by Clear(), GetSize(), GetValues(), InsertTail(), and RemoveHead().

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

- **RemoveNoise.hh**
- **RemoveNoise.cc**

7.79 clSortedArray::clXSortedArray Class Reference

Public Member Functions

- **clXSortedArray** (const char *cpErrorMsg, int iErrorCode=0) throw ()

7.79.1 Constructor & Destructor Documentation

7.79.1.1 clSortedArray::clXSortedArray::clXSortedArray (const char * *cpErrorMsg*, int *iErrorCode* = 0) throw () `[inline]`

Definition at line 50 of file RemoveNoise.hh.

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

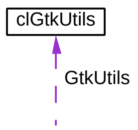
- **RemoveNoise.hh**

7.80 clSoundChGUI Class Reference

Per channel GUI for **clSoundUI**(p. 502).

`#include <SoundUI.hh>`

Collaboration diagram for clSoundChGUI:



Public Member Functions

- **clSoundChGUI** (GtkWidget ∗, int, int, long)
- ~**clSoundChGUI** ()

Public Attributes

- GList ∗ **glServers**
- GtkWidget ∗ **gwFrame**
- GtkWidget ∗ **gwVBox**
- GtkWidget ∗ **gwTable1**
- GtkWidget ∗ **gwLServer**
- GtkWidget ∗ **gwCServer**
- GtkWidget ∗ **gwLChannel**
- GtkObject ∗ **goAChannel**
- GtkWidget ∗ **gwSBChannel**
- GtkWidget ∗ **gwBConnect**
- GtkWidget ∗ **gwLInputLevel**
- GtkWidget ∗ **gwPBInputLevel**
- GtkWidget ∗ **gwCBEq**
- GtkWidget ∗ **gwTableEq**
- GtkWidget ∗ **gwLOutputLevel**
- GtkObject ∗ **goAOutputLevel**
- GtkWidget ∗ **gwVSOutputLevel**
- GtkWidget ∗ **gwPBOutputLevel**
- GtkWidget ∗ **gwaLEqLevel** [SUI_EQ_MAXOCTS]
- GtkObject ∗ **goaAEqLevel** [SUI_EQ_MAXOCTS]
- GtkWidget ∗ **gwaVSEqLevel** [SUI_EQ_MAXOCTS]
- GtkWidget ∗ **gwCurveEq**
- GtkWidget ∗ **gwBApplyCurve**

Private Attributes

- **clGtkUtils GtkUtils**

7.80.1 Detailed Description

Per channel GUI for **clSoundUI**(p. 502).

Definition at line 81 of file SoundUI.hh.

7.80.2 Constructor & Destructor Documentation

7.80.2.1 clSoundChGUI::clSoundChGUI (GtkWidget *, int, int, long)

Definition at line 1088 of file SoundUI.cc.

References clGtkUtils::ComboListFromFile(), cpBApplyCurveTxt, cpBConnectTxt, cpChannelPfx, cpLChannelTxt, cpLEqTxt, cpLInputLevelTxt, cpLOutputLevelTxt, cpLServerTxt, GDT, glServers, goaAEqLevel, goAChannel, goAOutputLevel, GtkUtils, gwaLEqLevel, gwaVSEqLevel, gwBApplyCurve, gwBConnect, gwCBEq, gwCServer, gwCurveEq, gwFrame, gwLChannel, gwLInputLevel, gwLOutputLevel, gwLServer, gwPBInputLevel, gwPBOutputLevel, gwSBChannel, gwTable1, gwTableEq, gwVBox, gwVSOutputLevel, SUI_CH_HIGHER, SUI_CH_LOWER, SUI_CH_VALUE, SUI_CONV_LEN, SUI_EQ_RANGE, SUI_EQ_STEP, SUI_HOSTFILE, SUI_IN_HIGHER, SUI_IN_LOWER, SUI_IN_VALUE, SUI_OUT_HIGHER, SUI_OUT_LOWER, SUI_OUT_RANGE, SUI_OUT_STEP, SUI_OUT_VALUE, SUI_PADDING, and SUI_SERV_MAXLEN.

7.80.2.2 clSoundChGUI::~clSoundChGUI ()

Definition at line 1273 of file SoundUI.cc.

7.80.3 Member Data Documentation

7.80.3.1 clGtkUtils clSoundChGUI::GtkUtils `[private]`

Definition at line 83 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.2 GList* clSoundChGUI::glServers

Definition at line 88 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.3 GtkWidget∗ clSoundChGUI::gwFrame

Definition at line 90 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.4 GtkWidget∗ clSoundChGUI::gwVBox

Definition at line 91 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.5 GtkWidget∗ clSoundChGUI::gwTable1

Definition at line 93 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.6 GtkWidget∗ clSoundChGUI::gwLServer

Definition at line 94 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.7 GtkWidget∗ clSoundChGUI::gwCServer

Definition at line 95 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.8 GtkWidget∗ clSoundChGUI::gwLChannel

Definition at line 96 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.9 GtkObject∗ clSoundChGUI::goAChannel

Definition at line 97 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.10 GtkWidget∗ clSoundChGUI::gwSBChannel

Definition at line 98 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.11 GtkWidget∗ clSoundChGUI::gwBConnect

Definition at line 99 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.12 GtkWidget∗ clSoundChGUI::gwLInputLevel

Definition at line 101 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.13 GtkWidget∗ clSoundChGUI::gwPBInputLevel

Definition at line 102 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.14 GtkWidget∗ clSoundChGUI::gwCBEq

Definition at line 103 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.15 GtkWidget∗ clSoundChGUI::gwTableEq

Definition at line 105 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.16 GtkWidget∗ clSoundChGUI::gwLOutputLevel

Definition at line 106 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.17 GtkObject∗ clSoundChGUI::goAOutputLevel

Definition at line 107 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.18 GtkWidget∗ clSoundChGUI::gwVSOutputLevel

Definition at line 108 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.19 GtkWidget* clSoundChGUI::gwPBOutputLevel

Definition at line 109 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.20 GtkWidget* clSoundChGUI::gwaLEqLevel[SUI_EQ_-MAXOCTS]

Definition at line 110 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.21 GtkObject* clSoundChGUI::goaAEqLevel[SUI_EQ_-MAXOCTS]

Definition at line 111 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.22 GtkWidget* clSoundChGUI::gwaVSEqLevel[SUI_EQ_-MAXOCTS]

Definition at line 112 of file SoundUI.hh.

Referenced by clSoundChGUI().

7.80.3.23 GtkWidget* clSoundChGUI::gwCurveEq

Definition at line 114 of file SoundUI.hh.

Referenced by clSoundChGUI(), and clSoundUI::OnMotionCurve().

7.80.3.24 GtkWidget* clSoundChGUI::gwBApplyCurve

Definition at line 115 of file SoundUI.hh.

Referenced by clSoundChGUI().

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

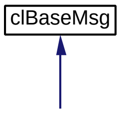
- **SoundUI.hh**
- **SoundUI.cc**

7.81 clSoundMsg Class Reference

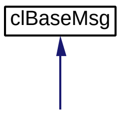
Sound server communication.

`#include <Messages.hh>`

Inheritance diagram for clSoundMsg:



Collaboration diagram for clSoundMsg:



Public Member Functions

- void **SetStart** (char ∗, const **stpSoundStart**)

  Start message from server.

- void **GetStart** (const char ∗, **stpSoundStart**)
- void **SetData** (void ∗, const struct timeval ∗, const float ∗, int)

  Datastream from server.

- void **SetData** (void ∗, const struct timeval ∗, const double ∗, int)
- void **SetData** (void ∗, const float ∗, int)
- void **SetData** (void ∗, const double ∗, int)
- int **GetData** (const void ∗, struct timeval ∗, float ∗)
- int **GetData** (const void ∗, struct timeval ∗, double ∗)
- void **GetData** (const void ∗, float ∗, int)
- void **GetData** (const void ∗, double ∗, int)

7.81.1 Detailed Description

Sound server communication.

Definition at line 455 of file Messages.hh.

7.81.2 Member Function Documentation

7.81.2.1 void clSoundMsg::SetStart (char *, const *stpSoundStart*)

Start message from server.

Parameters:

cpMsgBuf Message buffer

stpSoundStart Message

Definition at line 79 of file Messages.cc.

References _stSoundStart::dSampleRate, GLOBAL_HEADER_LEN, _stSoundStart::iChannels, _stSoundStart::iCompress, and _stSoundStart::iFragmentSize.

Referenced by clSoundSrv::ServeClientExec(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clFileSrv::ServeClientThread(), clComediSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), and clBeamSrvMaster::ServeClientThread().

7.81.2.2 void clSoundMsg::GetStart (const char *, stpSoundStart)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 97 of file Messages.cc.

References _stSoundStart::dSampleRate, _stSoundStart::iChannels, _stSoundStart::iCompress, and _stSoundStart::iFragmentSize.

Referenced by clStreamDist::AudioInThread(), main(), and clSoundUI::SoundInThread().

7.81.2.3 void clSoundMsg::SetData (void *, const struct timeval *, const float *, int)

Datastream from server.

Parameters:

vpMsgBuf Message buffer

spTimeStamp Time stamp

fpData Data

iDataLen Data length

Definition at line 107 of file Messages.cc.

References clBaseMsg::EndianConv(), and GLOBAL_HEADER_LEN.

Referenced by clSoundSrvA::AudioInThread(), clSoundSrv2::AudioInThread(), clComediSrv::AudioInThread(), clSoundSrv::InputExec(), clFileSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), and clBeamSrvMaster::ServeClientThread().

7.81.2.4 void clSoundMsg::SetData (void ∗, const struct timeval ∗, const double ∗, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 128 of file Messages.cc.

References clBaseMsg::EndianConv(), and GLOBAL_HEADER_LEN.

7.81.2.5 void clSoundMsg::SetData (void ∗, const float ∗, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 149 of file Messages.cc.

References clBaseMsg::EndianConv().

7.81.2.6 void clSoundMsg::SetData (void ∗, const double ∗, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 161 of file Messages.cc.

References clBaseMsg::EndianConv().

7.81.2.7 int clSoundMsg::GetData (const void ∗, struct timeval ∗, float ∗)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 173 of file Messages.cc.

References clBaseMsg::EndianConv(), and GLOBAL_HEADER_LEN.

Referenced by clStreamDist::AudioInThread(), main(), and clSoundUI::SoundInThread().

7.81.2.8 int clSoundMsg::GetData (const void ∗, struct timeval ∗, double ∗)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 196 of file Messages.cc.

References clBaseMsg::EndianConv(), and GLOBAL_HEADER_LEN.

7.81.2.9 void clSoundMsg::GetData (const void *, float *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 219 of file Messages.cc.

References clBaseMsg::EndianConv().

7.81.2.10 void clSoundMsg::GetData (const void *, double *, int)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Definition at line 231 of file Messages.cc.

References clBaseMsg::EndianConv().

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

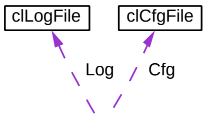
- **Messages.hh**
- **Messages.cc**

7.82 clSoundProxy Class Reference

Sound service proxy.

`#include <SoundProxy.hh>`

Collaboration diagram for clSoundProxy:



Public Member Functions

- **clSoundProxy** ()
- ~**clSoundProxy** ()
- int **Exec** ()
- void **Stop** ()
- void ∗ **SoundInThread** (void ∗)
- void ∗ **WaitConnectThread** (void ∗)
- void ∗ **ServeClientThread** (void ∗)

Private Member Functions

- void **AddToLog** (char, const char ∗)
- void **AddToLog** (char, const char ∗, int)
- int **FindFreeSlot** ()

Private Attributes

- volatile bool **bRun**
- bool **bServeClient** [SP_MAXCLIENTS]
- int **iServerPort**
- int **iServicePort**
- int **iClientSockH** [SP_MAXCLIENTS]
- char **cpLogFile** [_POSIX_PATH_MAX+1]
- char **cpServerHost** [SP_SERV_MAXLEN+1]
- char ∗ **cpFirstMsg**
- char ∗ **cpDataMsg**
- pthread_t **ptidSoundIn**
- pthread_t **ptidWaitConnect**
- pthread_t **ptidServeClient** [SP_MAXCLIENTS]
- clAlloc **FirstMsg**
- clAlloc **DataMsg**

- **clCfgFile Cfg**
- clCondition **CondData**
- **clLogFile Log**
- clMutex **MutexData**
- clMutex **MutexClass**

7.82.1 Detailed Description

Sound service proxy.

Note:
See soundsrv2 for architecture description.

Definition at line 49 of file SoundProxy.hh.

7.82.2 Constructor & Destructor Documentation

7.82.2.1 clSoundProxy::clSoundProxy ()

Definition at line 144 of file SoundProxy.cc.

References bRun, bServeClient, Cfg, cpDataMsg, cpFirstMsg, cpLogFile, cpServerHost, DataMsg, FirstMsg, clCfgFile::GetInt(), clCfgFile::GetStr(), GLOBAL_HEADER_LEN, iServerPort, iServicePort, Log, clLogFile::Open(), clCfgFile::SetFileName(), SP_BUFFER_SIZE, SP_CFGFILE, SP_DEF_LOGFILE, and SP_MAXCLIENTS.

7.82.2.2 clSoundProxy::∼clSoundProxy ()

Definition at line 164 of file SoundProxy.cc.

7.82.3 Member Function Documentation

7.82.3.1 void clSoundProxy::AddToLog (char, const char ∗) `[inline, private]`

Definition at line 108 of file SoundProxy.cc.

References clLogFile::Add(), Log, and MutexClass.

Referenced by Exec(), ServeClientThread(), SoundInThread(), and WaitConnectThread().

7.82.3.2 void clSoundProxy::AddToLog (char, const char ∗, int) `[inline, private]`

Definition at line 116 of file SoundProxy.cc.

References clLogFile::Add(), Log, and MutexClass.

7.82.3.3 int clSoundProxy::FindFreeSlot () [private]

Definition at line 125 of file SoundProxy.cc.

References bServeClient, MutexClass, and SP_MAXCLIENTS.

Referenced by WaitConnectThread().

7.82.3.4 int clSoundProxy::Exec ()

Definition at line 169 of file SoundProxy.cc.

References AddToLog(), ptidSoundIn, ptidWaitConnect, Stop(), WrapSoundInThread(), and WrapWaitConnectThread().

Referenced by main().

7.82.3.5 void clSoundProxy::Stop ()

Definition at line 206 of file SoundProxy.cc.

References bRun, and MutexClass.

Referenced by Exec(), and SoundInThread().

7.82.3.6 void * clSoundProxy::SoundInThread (void *)

Definition at line 214 of file SoundProxy.cc.

References AddToLog(), bRun, bRun, clSockOp::Close(), CondData, clSockClie::Connect(), cpDataMsg, cpFirstMsg, cpServerHost, clSockOp::DisableNagle(), clSockOp::GetErrno(), clSockClie::GetErrno(), GLOBAL_HEADER_LEN, iServerPort, MutexClass, MutexData, clSockOp::ReadN(), clSockOp::ReadSelect(), clSockOp::SetHandle(), clSockOp::SetTypeOfService(), clSockOp::Shutdown(), SOp, SP_1ST_MSG_TIMEOUT, SP_BUFFER_SIZE, SP_CONV_BUF_LEN, SP_MSG_TIMEOUT, SP_SCHED_PRIORITY, and Stop().

Referenced by WrapSoundInThread().

7.82.3.7 void * clSoundProxy::WaitConnectThread (void *)

Definition at line 301 of file SoundProxy.cc.

References AddToLog(), clSockServ::Bind(), bRun, bServeClient, FindFreeSlot(), iClientSockH, iServicePort, MutexClass, ptidServeClient, SP_WAIT_CONN_TIMEOUT, clSockServ::WaitForConnect(), and WrapServeClientThread().

Referenced by WrapWaitConnectThread().

7.82.3.8 void * clSoundProxy::ServeClientThread (void *)

Definition at line 339 of file SoundProxy.cc.

References AddToLog(), bRun, clSockOp::Close(), CondData, cpDataMsg, cpFirstMsg, clSockOp::DisableNagle(), clSockOp::GetErrno(), clSockOp::GetPeerName(), GLOBAL_HEADER_LEN, iClientSockH, MutexClass, MutexData, clSockOp::SetHandle(), clSockOp::SetTypeOfService(), SOp, SP_1ST_MSG_TIMEOUT, SP_BUFFER_SIZE, SP_CONV_BUF_LEN, SP_MSG_TIMEOUT, SP_SCHED_PRIORITY, clSockOp::WriteN(), and clSockOp::WriteSelect().

Referenced by WrapServeClientThread().

7.82.4 Member Data Documentation

7.82.4.1 volatile bool clSoundProxy::bRun [private]

Definition at line 51 of file SoundProxy.hh.

Referenced by clSoundProxy(), ServeClientThread(), SoundInThread(), Stop(), and WaitConnectThread().

7.82.4.2 bool clSoundProxy::bServeClient[SP_MAXCLIENTS] [private]

Definition at line 52 of file SoundProxy.hh.

Referenced by clSoundProxy(), FindFreeSlot(), and WaitConnectThread().

7.82.4.3 int clSoundProxy::iServerPort [private]

Definition at line 53 of file SoundProxy.hh.

Referenced by clSoundProxy(), and SoundInThread().

7.82.4.4 int clSoundProxy::iServicePort [private]

Definition at line 54 of file SoundProxy.hh.

Referenced by clSoundProxy(), and WaitConnectThread().

7.82.4.5 int clSoundProxy::iClientSockH[SP_MAXCLIENTS] [private]

Definition at line 55 of file SoundProxy.hh.

Referenced by ServeClientThread(), and WaitConnectThread().

7.82.4.6 char clSoundProxy::cpLogFile[_POSIX_PATH_MAX + 1] [private]

Definition at line 56 of file SoundProxy.hh.

Referenced by clSoundProxy().

7.82.4.7 char clSoundProxy::cpServerHost[SP_SERV_MAXLEN + 1] [private]

Definition at line 57 of file SoundProxy.hh.

Referenced by clSoundProxy(), and SoundInThread().

7.82.4.8 char* clSoundProxy::cpFirstMsg [private]

Definition at line 58 of file SoundProxy.hh.

Referenced by clSoundProxy(), ServeClientThread(), and SoundInThread().

7.82.4.9 char* clSoundProxy::cpDataMsg [private]

Definition at line 59 of file SoundProxy.hh.

Referenced by clSoundProxy(), ServeClientThread(), and SoundInThread().

7.82.4.10 pthread_t clSoundProxy::ptidSoundIn [private]

Definition at line 60 of file SoundProxy.hh.

Referenced by Exec().

7.82.4.11 pthread_t clSoundProxy::ptidWaitConnect [private]

Definition at line 61 of file SoundProxy.hh.

Referenced by Exec().

7.82.4.12 pthread_t clSoundProxy::ptidServeClient[SP_-MAXCLIENTS] [private]

Definition at line 62 of file SoundProxy.hh.

Referenced by WaitConnectThread().

7.82.4.13 clAlloc clSoundProxy::FirstMsg [private]

Definition at line 63 of file SoundProxy.hh.

Referenced by clSoundProxy().

7.82.4.14 clAlloc clSoundProxy::DataMsg [private]

Definition at line 64 of file SoundProxy.hh.

Referenced by clSoundProxy().

7.82.4.15 clCfgFile clSoundProxy::Cfg [private]

Definition at line 65 of file SoundProxy.hh.

Referenced by clSoundProxy().

7.82.4.16 clCondition clSoundProxy::CondData [private]

Definition at line 66 of file SoundProxy.hh.

Referenced by ServeClientThread(), and SoundInThread().

7.82.4.17 clLogFile clSoundProxy::Log [private]

Definition at line 67 of file SoundProxy.hh.

Referenced by AddToLog(), and clSoundProxy().

7.82.4.18 clMutex clSoundProxy::MutexData [private]

Definition at line 68 of file SoundProxy.hh.

Referenced by ServeClientThread(), and SoundInThread().

7.82.4.19 clMutex clSoundProxy::MutexClass [private]

Definition at line 69 of file SoundProxy.hh.

Referenced by AddToLog(), FindFreeSlot(), ServeClientThread(), SoundInThread(), Stop(), and WaitConnectThread().

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

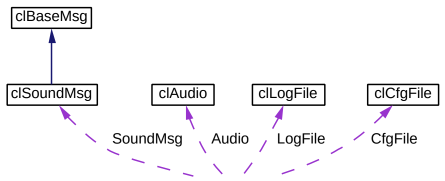
- **SoundProxy.hh**
- **SoundProxy.cc**

7.83 clSoundSrv Class Reference

Sound card input server.

`#include <SoundSrv.hh>`

Collaboration diagram for clSoundSrv:



Public Member Functions

- **clSoundSrv** (const char ∗, const char ∗)
- ∼**clSoundSrv** ()
- void **Exec** ()
- void **Quit** ()
- void **Quit** (int)
- void **InputExec** ()
- void **WaitConnectExec** ()
- void **ServeClientExec** (void ∗)
- bool **Run** ()

Private Member Functions

- int **FindFreeClient** ()
- int **FindThisHandle** (int)

Private Attributes

- int **iAudioFrmt**
- int **iAudioSr**
- int **iAudioCh**
- int **iAudioTypeSize**
- int **iErrorCount**
- int **iClientIdx**
- int **ipClientH** [SS_MAXCLIENTS]
- int **iSampleCount**
- int **iOutBufSize**
- bool **bRun**
- char **cpProgName** [_POSIX_PATH_MAX]

- char * **cpOutBuf**
- sigset_t **sigsetQuit**
- pth_t **tidInput**
- pth_t **tidWaitConnect**
- pth_t **ptidServeClient** [SS_MAXCLIENTS]
- clPthMutex **MutexThis**
- clPthCond **CondDataAvail**
- **clSoundMsg SoundMsg**
- clDSPOp **DSP**
- **clCfgFile * CfgFile**
- **clLogFile * LogFile**
- **clAudio * Audio**

7.83.1 Detailed Description

Sound card input server.

This uses great GNU Pth library for threads. This seems to be more efficient in some cases than soundsrv2 due to nonpre-emptive behaviour of used threading library. It, however, requires select() support from soundcard which, unfortunately, all soundcard drivers don't support...

Definition at line 54 of file SoundSrv.hh.

7.83.2 Constructor & Destructor Documentation

7.83.2.1 clSoundSrv::clSoundSrv (const char *, const char *)

Definition at line 145 of file SoundSrv.cc.

References clLogFile::Add(), Audio, AUDIO_READ, bRun, CfgFile, cpOutBuf, cpProgName, GDT, clAudio::GetError(), clAudio::GetFragmentSize(), clCfgFile::GetInt(), clAudio::GetVersion(), iAudioCh, iAudioFrmt, iAudioSr, iAudioTypeSize, iOutBufSize, ipClientH, iSampleCount, LogFile, SS_LOGFILE, SS_MAXCLIENTS, SS_PTH_MAJ, SS_PTH_MIN, SS_PTH_PL, SS_SND_CHANNELS, SS_SND_FORMAT, SS_SND_FORMAT_SIZE, and SS_SND_SAMPLERATE.

7.83.2.2 clSoundSrv::~clSoundSrv ()

Definition at line 241 of file SoundSrv.cc.

References clLogFile::Add(), Audio, CfgFile, cpOutBuf, iOutBufSize, and LogFile.

7.83.3 Member Function Documentation

7.83.3.1 int clSoundSrv::FindFreeClient () [inline, private]

Definition at line 651 of file SoundSrv.cc.

References ipClientH, and SS_MAXCLIENTS.

Referenced by WaitConnectExec().

7.83.3.2 int clSoundSrv::FindThisHandle (int) [inline, private]

Definition at line 666 of file SoundSrv.cc.

References ipClientH, and SS_MAXCLIENTS.

Referenced by ServeClientExec().

7.83.3.3 void clSoundSrv::Exec ()

Definition at line 252 of file SoundSrv.cc.

References clLogFile::Add(), Input(), LogFile, Quit(), sigsetQuit, tidInput, tidWaitConnect, and WaitConnect().

Referenced by main().

7.83.3.4 void clSoundSrv::Quit ()

Definition at line 298 of file SoundSrv.cc.

References bRun.

Referenced by Exec(), and InputExec().

7.83.3.5 void clSoundSrv::Quit (int)

Definition at line 304 of file SoundSrv.cc.

References clLogFile::Add(), bRun, and LogFile.

7.83.3.6 void clSoundSrv::InputExec ()

Definition at line 321 of file SoundSrv.cc.

References clLogFile::Add(), Audio, bRun, CondDataAvail, cpOutBuf, DSP, GDT, clAudio::GetHandle(), iAudioTypeSize, iOutBufSize, iSampleCount, LogFile, MutexThis, Quit(), clSoundMsg::SetData(), SoundMsg, SoundSrv, SS_MAXERRORS, and SS_SHUTDOWNFILE.

Referenced by Input().

7.83.3.7 void clSoundSrv::WaitConnectExec ()

Definition at line 411 of file SoundSrv.cc.

References clLogFile::Add(), bRun, CfgFile, cpProgName, FindFreeClient(), clCfgFile::GetInt(), iClientIdx, ipClientH, LogFile, ptidServeClient, ServeClient(), SS_DEFAULT_PORT, SS_MAXCLIENTS, and tidInput.

Referenced by WaitConnect().

7.83.3.8 void clSoundSrv::ServeClientExec (void *)

Definition at line 510 of file SoundSrv.cc.

References clLogFile::Add(), bRun, CondDataAvail, cpOutBuf, _stSoundStart::dSampleRate, FindThisHandle(), GLOBAL_HEADER_LEN, iAudioCh, iAudioSr, _stSoundStart::iChannels, _stSoundStart::iFragmentSize, iOutBufSize, ipClientH, iSampleCount, LogFile, MutexThis, clSoundMsg::SetStart(), SoundMsg, SS_SOCKET_BUF_FRAGS, and tidInput.

Referenced by ServeClient().

7.83.3.9 bool clSoundSrv::Run () `[inline]`

Definition at line 91 of file SoundSrv.hh.

References bRun.

7.83.4 Member Data Documentation

7.83.4.1 int clSoundSrv::iAudioFrmt `[private]`

Definition at line 56 of file SoundSrv.hh.

Referenced by clSoundSrv().

7.83.4.2 int clSoundSrv::iAudioSr `[private]`

Definition at line 57 of file SoundSrv.hh.

Referenced by clSoundSrv(), and ServeClientExec().

7.83.4.3 int clSoundSrv::iAudioCh `[private]`

Definition at line 58 of file SoundSrv.hh.

Referenced by clSoundSrv(), and ServeClientExec().

7.83.4.4 int clSoundSrv::iAudioTypeSize [private]

Definition at line 59 of file SoundSrv.hh.

Referenced by clSoundSrv(), and InputExec().

7.83.4.5 int clSoundSrv::iErrorCount [private]

Definition at line 60 of file SoundSrv.hh.

7.83.4.6 int clSoundSrv::iClientIdx [private]

Definition at line 61 of file SoundSrv.hh.

Referenced by WaitConnectExec().

7.83.4.7 int clSoundSrv::ipClientH[SS_MAXCLIENTS] [private]

Definition at line 62 of file SoundSrv.hh.

Referenced by clSoundSrv(), FindFreeClient(), FindThisHandle(), ServeClient-Exec(), and WaitConnectExec().

7.83.4.8 int clSoundSrv::iSampleCount [private]

Definition at line 63 of file SoundSrv.hh.

Referenced by clSoundSrv(), InputExec(), and ServeClientExec().

7.83.4.9 int clSoundSrv::iOutBufSize [private]

Definition at line 64 of file SoundSrv.hh.

Referenced by clSoundSrv(), InputExec(), ServeClientExec(), and ~clSound-Srv().

7.83.4.10 bool clSoundSrv::bRun [private]

Definition at line 65 of file SoundSrv.hh.

Referenced by clSoundSrv(), InputExec(), Quit(), Run(), ServeClientExec(), and WaitConnectExec().

7.83.4.11 char clSoundSrv::cpProgName[_POSIX_PATH_MAX] [private]

Definition at line 66 of file SoundSrv.hh.

Referenced by clSoundSrv(), and WaitConnectExec().

7.83.4.12 char* clSoundSrv::cpOutBuf [private]

Definition at line 67 of file SoundSrv.hh.

Referenced by clSoundSrv(), InputExec(), ServeClientExec(), and ~clSoundSrv().

7.83.4.13 sigset_t clSoundSrv::sigsetQuit [private]

Definition at line 68 of file SoundSrv.hh.

Referenced by Exec().

7.83.4.14 pth_t clSoundSrv::tidInput [private]

Definition at line 69 of file SoundSrv.hh.

Referenced by Exec(), ServeClientExec(), and WaitConnectExec().

7.83.4.15 pth_t clSoundSrv::tidWaitConnect [private]

Definition at line 70 of file SoundSrv.hh.

Referenced by Exec().

7.83.4.16 pth_t clSoundSrv::ptidServeClient[SS_MAXCLIENTS] [private]

Definition at line 71 of file SoundSrv.hh.

Referenced by WaitConnectExec().

7.83.4.17 clPthMutex clSoundSrv::MutexThis [private]

Definition at line 72 of file SoundSrv.hh.

Referenced by InputExec(), and ServeClientExec().

7.83.4.18 clPthCond clSoundSrv::CondDataAvail [private]

Definition at line 73 of file SoundSrv.hh.

Referenced by InputExec(), and ServeClientExec().

7.83.4.19 clSoundMsg clSoundSrv::SoundMsg [private]

Definition at line 74 of file SoundSrv.hh.

Referenced by InputExec(), and ServeClientExec().

7.83.4.20 clDSPOp clSoundSrv::DSP [private]

Definition at line 75 of file SoundSrv.hh.

Referenced by InputExec().

7.83.4.21 clCfgFile* clSoundSrv::CfgFile [private]

Definition at line 76 of file SoundSrv.hh.

Referenced by clSoundSrv(), WaitConnectExec(), and ~clSoundSrv().

7.83.4.22 clLogFile* clSoundSrv::LogFile [private]

Definition at line 77 of file SoundSrv.hh.

Referenced by clSoundSrv(), Exec(), InputExec(), Quit(), ServeClientExec(), WaitConnectExec(), and ~clSoundSrv().

7.83.4.23 clAudio* clSoundSrv::Audio [private]

Definition at line 78 of file SoundSrv.hh.

Referenced by clSoundSrv(), InputExec(), and ~clSoundSrv().

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

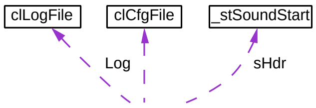
- **SoundSrv.hh**
- **SoundSrv.cc**

7.84 clSoundSrv2 Class Reference

SoundServer2.

`#include <SoundSrv2.hh>`

Collaboration diagram for clSoundSrv2:



Public Member Functions

- **clSoundSrv2** ()
- ~**clSoundSrv2** ()
- int **Main** (int, char ∗∗)
- void ∗ **AudioInThread** (void ∗)
- void ∗ **ServeClientThread** (void ∗)
- void **Stop** ()

Public Attributes

- **clLogFile Log**

Private Member Functions

- bool **GetAudioCfg** (char ∗, int ∗, int ∗, int ∗, int ∗)
- bool **InitCompress** (int, int, int, int, int)

Private Attributes

- volatile bool **bRun**
- volatile int **iAudioBufSize**
- volatile int **iBlockCntr**
- **stSoundStart sHdr**
- clAlloc **AudioBuf**
- clAlloc **CompHead**
- clAlloc **FLACFrame**
- clMutex **MtxAudio**
- clCondition **CndAudio**
- **clCfgFile Cfg**

7.84.1 Detailed Description

SoundServer2.

This uses freely spinning architecture based on condition variables. When thread gets buffer read from the soundcard it copies it's contents to shared buffer and broadcasts condition. Then all client servicing threads (which are waiting for condition) copy it's contents to private buffers and start sending out. Thus, if one of client servicing threads get stuck on blocking socket it doesn't take any CPU time and just misses n blocks of data.

Soundproxy and streamdist also use this architecture. This ensures, that one client can't affect others performance.

Definition at line 67 of file SoundSrv2.hh.

7.84.2 Constructor & Destructor Documentation

7.84.2.1 clSoundSrv2::clSoundSrv2 ()

Definition at line 268 of file SoundSrv2.cc.

References clLogFile::Add(), bRun, Cfg, iAudioBufSize, iBlockCntr, Log, clLogFile::Open(), clCfgFile::SetFileName(), SS2_CFGFILE, and SS2_LOGFILE.

7.84.2.2 clSoundSrv2::~clSoundSrv2 ()

Definition at line 282 of file SoundSrv2.cc.

References clLogFile::Add(), and Log.

7.84.3 Member Function Documentation

7.84.3.1 bool clSoundSrv2::GetAudioCfg (char *, int *, int *, int *, int *) `[private]`

Definition at line 112 of file SoundSrv2.cc.

References clLogFile::Add(), Cfg, clCfgFile::GetInt(), clCfgFile::GetStr(), and Log.

Referenced by AudioInThread().

7.84.3.2 bool clSoundSrv2::InitCompress (int, int, int, int, int) `[private]`

Definition at line 144 of file SoundSrv2.cc.

References clLogFile::Add(), Cfg, FLACFrame, clCfgFile::GetInt(), Log, and MSG_SOUND_COMPRESS_FLAC.

Referenced by AudioInThread().

7.84.3.3 int clSoundSrv2::Main (int, char **)

Definition at line 295 of file SoundSrv2.cc.

References clLogFile::Add(), bDaemon, clSockServ::Bind(), bRun, Cfg, clCfgFile::GetInt(), GLOBAL_VERSMAJ, GLOBAL_VERSMIN, GLOBAL_VERSPL, Log, SigHandler(), SS2_CONNECT_TIMEOUT, SS2_-SHUTDOWNFILE, Stop(), clSockServ::WaitForConnect(), WrapAudioIn-Thread(), and WrapServeClientThread().

Referenced by main().

7.84.3.4 void * clSoundSrv2::AudioInThread (void *)

Definition at line 371 of file SoundSrv2.cc.

References clLogFile::Add(), AUDIO_READ, AudioBuf, bRun, Cfg, Cnd-Audio, _stSoundStart::dSampleRate, FLACFrame, GDT, GetAudioCfg(), cl-Audio::GetFragmentSize(), clCfgFile::GetInt(), clAudio::GetVersion(), iAudio-BufSize, iBlockCntr, _stSoundStart::iChannels, _stSoundStart::iCompress, _st-SoundStart::iFragmentSize, InitCompress(), Log, MSG_SOUND_COMPRESS_-FLAC, MSG_SOUND_COMPRESS_NONE, MtxAudio, clAudio::Open(), cl-Audio::Read(), clSoundMsg::SetData(), clAudio::SetFragment(), sHdr, SS2_-FRAG_SIZE_DEFAULT, SS2_INTHREAD_PRIORITY, SS2_LOGENTRY_-SIZE, SS2_OSS_MAJOR, SS2_OSS_MINOR, SS2_OSS_PL, and Stop().

Referenced by WrapAudioInThread().

7.84.3.5 void * clSoundSrv2::ServeClientThread (void *)

Definition at line 570 of file SoundSrv2.cc.

References clLogFile::Add(), AudioBuf, bRun, Cfg, CndAudio, Comp-Head, clSockOp::DisableNagle(), clCfgFile::GetInt(), clSockOp::GetPeer-Name(), GLOBAL_HEADER_LEN, iAudioBufSize, iBlockCntr, _stSound-Start::iCompress, Log, MtxAudio, clSockOp::SetSendBufSize(), clSound-Msg::SetStart(), clSockOp::SetTypeOfService(), sHdr, SOp, SS2_LOGENTRY_-SIZE, SS2_OUTTHREAD_PRIORITY, SS2_SOCKET_BUF_FRAGS, and cl-SockOp::WriteN().

Referenced by WrapServeClientThread().

7.84.3.6 void clSoundSrv2::Stop () `[inline]`

Definition at line 97 of file SoundSrv2.hh.

References bRun.

Referenced by AudioInThread(), Main(), and SigHandler().

7.84.4 Member Data Documentation

7.84.4.1 volatile bool clSoundSrv2::bRun [private]

Definition at line 69 of file SoundSrv2.hh.

Referenced by AudioInThread(), clSoundSrv2(), Main(), ServeClientThread(), and Stop().

7.84.4.2 volatile int clSoundSrv2::iAudioBufSize [private]

Definition at line 70 of file SoundSrv2.hh.

Referenced by AudioInThread(), clSoundSrv2(), and ServeClientThread().

7.84.4.3 volatile int clSoundSrv2::iBlockCntr [private]

Definition at line 71 of file SoundSrv2.hh.

Referenced by AudioInThread(), clSoundSrv2(), and ServeClientThread().

7.84.4.4 stSoundStart clSoundSrv2::sHdr [private]

Definition at line 72 of file SoundSrv2.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.84.4.5 clAlloc clSoundSrv2::AudioBuf [private]

Definition at line 76 of file SoundSrv2.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.84.4.6 clAlloc clSoundSrv2::CompHead [private]

Definition at line 77 of file SoundSrv2.hh.

Referenced by ServeClientThread().

7.84.4.7 clAlloc clSoundSrv2::FLACFrame [private]

Definition at line 78 of file SoundSrv2.hh.

Referenced by AudioInThread(), and InitCompress().

7.84.4.8 clMutex clSoundSrv2::MtxAudio [private]

Definition at line 79 of file SoundSrv2.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.84.4.9 clCondition clSoundSrv2::CndAudio [private]

Definition at line 83 of file SoundSrv2.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.84.4.10 clCfgFile clSoundSrv2::Cfg [private]

Definition at line 84 of file SoundSrv2.hh.

Referenced by AudioInThread(), clSoundSrv2(), GetAudioCfg(), InitCompress(), Main(), and ServeClientThread().

7.84.4.11 clLogFile clSoundSrv2::Log

Definition at line 91 of file SoundSrv2.hh.

Referenced by AudioInThread(), clSoundSrv2(), GetAudioCfg(), InitCompress(), Main(), ServeClientThread(), SigHandler(), and ~clSoundSrv2().

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

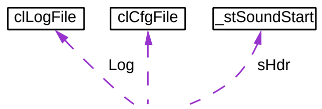
- **SoundSrv2.hh**
- **SoundSrv2.cc**

7.85 clSoundSrvA Class Reference

SoundServer for ALSA.

`#include <SoundSrvA.hh>`

Collaboration diagram for clSoundSrvA:



Public Member Functions

- **clSoundSrvA** ()
- ~**clSoundSrvA** ()
- int **Main** (int, char ∗∗)
- void ∗ **AudioInThread** (void ∗)
- void ∗ **ServeClientThread** (void ∗)
- void **Stop** ()

Public Attributes

- **clLogFile Log**

Private Member Functions

- bool **GetAudioCfg** (int ∗, int ∗, int ∗, int ∗, int ∗, int ∗, int ∗, int ∗)
- bool **InitCompress** (int, int, int, int, int)

Private Attributes

- volatile bool **bRun**
- volatile int **iAudioBufSize**
- volatile int **iBlockCntr**
- **stSoundStart sHdr**
- clAlloc **AudioBuf**
- clAlloc **CompHead**
- clAlloc **FLACFrame**
- clMutex **MtxAudio**
- clCondition **CndAudio**
- **clCfgFile Cfg**

7.85.1 Detailed Description

SoundServer for ALSA.

Note:
See soundsrv2 for architecture description.

Definition at line 58 of file SoundSrvA.hh.

7.85.2 Constructor & Destructor Documentation

7.85.2.1 clSoundSrvA::clSoundSrvA ()

Definition at line 276 of file SoundSrvA.cc.

References clLogFile::Add(), bRun, Cfg, iAudioBufSize, iBlockCntr, Log, clLogFile::Open(), clCfgFile::SetFileName(), SSA_CFGFILE, and SSA_LOGFILE.

7.85.2.2 clSoundSrvA::~clSoundSrvA ()

Definition at line 290 of file SoundSrvA.cc.

References clLogFile::Add(), and Log.

7.85.3 Member Function Documentation

7.85.3.1 bool clSoundSrvA::GetAudioCfg (int *, int *, int *, int *, int *, int *, int *, int *) `[private]`

Definition at line 110 of file SoundSrvA.cc.

References clLogFile::Add(), Cfg, clCfgFile::GetInt(), Log, and SSA_FRAG_SIZE_DEFAULT.

Referenced by AudioInThread().

7.85.3.2 bool clSoundSrvA::InitCompress (int, int, int, int, int) `[private]`

Definition at line 152 of file SoundSrvA.cc.

References clLogFile::Add(), Cfg, FLACFrame, clCfgFile::GetInt(), Log, and MSG_SOUND_COMPRESS_FLAC.

Referenced by AudioInThread().

7.85.3.3 int clSoundSrvA::Main (int, char **)

Definition at line 303 of file SoundSrvA.cc.

References clLogFile::Add(), bDaemon, clSockServ::Bind(), bRun, Cfg, clCfgFile::GetInt(), GLOBAL_VERSMAJ, GLOBAL_VERSMIN, GLOBAL_VERSPL, Log, SigHandler(), SSA_CONNECT_TIMEOUT, SSA_SHUTDOWNFILE, Stop(), clSockServ::WaitForConnect(), WrapAudioInThread(), and WrapServeClientThread().

Referenced by main().

7.85.3.4 void * clSoundSrvA::AudioInThread (void *)

Definition at line 379 of file SoundSrvA.cc.

References clLogFile::Add(), AudioBuf, bRun, Cfg, CndAudio, _stSoundStart::dSampleRate, FLACFrame, GDT, GetAudioCfg(), clCfgFile::GetInt(), iAudioBufSize, iBlockCntr, _stSoundStart::iChannels, _stSoundStart::iCompress, _stSoundStart::iFragmentSize, InitCompress(), Log, MSG_SOUND_COMPRESS_FLAC, MSG_SOUND_COMPRESS_NONE, MtxAudio, clSoundMsg::SetData(), sHdr, SSA_FRAG_SIZE_DEFAULT, SSA_INTHREAD_PRIORITY, SSA_LOGENTRY_SIZE, and Stop().

Referenced by WrapAudioInThread().

7.85.3.5 void * clSoundSrvA::ServeClientThread (void *)

Definition at line 664 of file SoundSrvA.cc.

References clLogFile::Add(), AudioBuf, bRun, Cfg, CndAudio, CompHead, clSockOp::DisableNagle(), clCfgFile::GetInt(), clSockOp::GetPeerName(), GLOBAL_HEADER_LEN, iAudioBufSize, iBlockCntr, _stSoundStart::iCompress, Log, MtxAudio, clSockOp::SetSendBufSize(), clSoundMsg::SetStart(), clSockOp::SetTypeOfService(), sHdr, SOp, SSA_LOGENTRY_SIZE, SSA_OUTTHREAD_PRIORITY, SSA_SOCKET_BUF_FRAGS, and clSockOp::WriteN().

Referenced by WrapServeClientThread().

7.85.3.6 void clSoundSrvA::Stop () `[inline]`

Definition at line 89 of file SoundSrvA.hh.

References bRun.

Referenced by AudioInThread(), Main(), and SigHandler().

7.85.4 Member Data Documentation

7.85.4.1 volatile bool clSoundSrvA::bRun `[private]`

Definition at line 60 of file SoundSrvA.hh.

Referenced by AudioInThread(), clSoundSrvA(), Main(), ServeClientThread(), and Stop().

7.85.4.2 volatile int clSoundSrvA::iAudioBufSize [private]

Definition at line 61 of file SoundSrvA.hh.

Referenced by AudioInThread(), clSoundSrvA(), and ServeClientThread().

7.85.4.3 volatile int clSoundSrvA::iBlockCntr [private]

Definition at line 62 of file SoundSrvA.hh.

Referenced by AudioInThread(), clSoundSrvA(), and ServeClientThread().

7.85.4.4 stSoundStart clSoundSrvA::sHdr [private]

Definition at line 63 of file SoundSrvA.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.85.4.5 clAlloc clSoundSrvA::AudioBuf [private]

Definition at line 67 of file SoundSrvA.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.85.4.6 clAlloc clSoundSrvA::CompHead [private]

Definition at line 68 of file SoundSrvA.hh.

Referenced by ServeClientThread().

7.85.4.7 clAlloc clSoundSrvA::FLACFrame [private]

Definition at line 69 of file SoundSrvA.hh.

Referenced by AudioInThread(), and InitCompress().

7.85.4.8 clMutex clSoundSrvA::MtxAudio [private]

Definition at line 70 of file SoundSrvA.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.85.4.9 clCondition clSoundSrvA::CndAudio [private]

Definition at line 74 of file SoundSrvA.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.85.4.10 clCfgFile clSoundSrvA::Cfg [private]

Definition at line 75 of file SoundSrvA.hh.

Referenced by AudioInThread(), clSoundSrvA(), GetAudioCfg(), InitCompress(), Main(), and ServeClientThread().

7.85.4.11 clLogFile clSoundSrvA::Log

Definition at line 83 of file SoundSrvA.hh.

Referenced by AudioInThread(), clSoundSrvA(), GetAudioCfg(), InitCompress(), Main(), ServeClientThread(), SigHandler(), and ~clSoundSrvA().

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

- **SoundSrvA.hh**
- **SoundSrvA.cc**

7.86 _stSoundStart Struct Reference

Header message from soundsrv.

`#include <Messages.hh>`

Public Attributes

- int **iChannels**

  Number of channels.

- double **dSampleRate**

  Samplerate.

- int **iFragmentSize**

  Fragment size (in samples).

- int **iCompress**

  Compression.

7.86.1 Detailed Description

Header message from soundsrv.

Definition at line 199 of file Messages.hh.

7.86.2 Member Data Documentation

7.86.2.1 int _stSoundStart::iChannels

Number of channels.

Definition at line 201 of file Messages.hh.

Referenced by clStreamDist::AudioInThread(), clSoundSrvA::AudioInThread(), clSoundSrv2::AudioInThread(), clComediSrv::AudioInThread(), clSoundMsg::GetStart(), main(), clSoundSrv::ServeClientExec(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), clSoundMsg::SetStart(), and clSoundUI::SoundInThread().

7.86.2.2 double _stSoundStart::dSampleRate

Samplerate.

Definition at line 202 of file Messages.hh.

Referenced by clStreamDist::AudioInThread(), clSoundSrvA::AudioInThread(), clSoundSrv2::AudioInThread(), clComediSrv::AudioInThread(), clSoundMsg::GetStart(), main(), clSoundSrv::ServeClientExec(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), clSoundMsg::SetStart(), and clSoundUI::SoundInThread().

7.86.2.3 int _stSoundStart::iFragmentSize

Fragment size (in samples).

Definition at line 203 of file Messages.hh.

Referenced by clStreamDist::AudioInThread(), clSoundSrvA::AudioInThread(), clSoundSrv2::AudioInThread(), clComediSrv::AudioInThread(), clSoundMsg::GetStart(), clSoundSrv::ServeClientExec(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), and clSoundMsg::SetStart().

7.86.2.4 int _stSoundStart::iCompress

Compression.

Definition at line 204 of file Messages.hh.

Referenced by clStreamDist::AudioInThread(), clSoundSrvA::AudioInThread(), clSoundSrv2::AudioInThread(), clComediSrv::AudioInThread(), clSoundMsg::GetStart(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clComediSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), and clSoundMsg::SetStart().

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

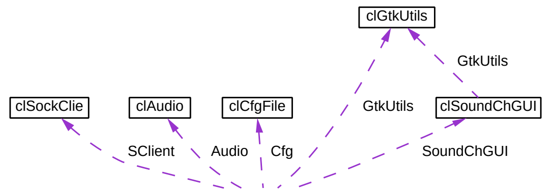
- **Messages.hh**

7.87 clSoundUI Class Reference

Sound user interface.

`#include <SoundUI.hh>`

Collaboration diagram for clSoundUI:



Public Member Functions

- **clSoundUI** (int ∗, char ∗∗∗)
- ∼**clSoundUI** ()
- int **Exec** ()
- gboolean **OnDeleteEvent** (GtkWidget ∗gwSender, GdkEvent ∗geEvent, gpointer gpData)
- void **OnClickedEvent** (GtkButton ∗, gpointer)
- gint **OnTimeoutEvent** (gpointer)
- void **OnToggledEvent** (GtkToggleButton ∗, gpointer)
- void **OnValueChangedEvent** (GtkAdjustment ∗, gpointer)
- void **OnApplyCurveClicked** (GtkButton ∗, gpointer)
- void **OnMotionCurve** (GtkWidget ∗, GdkEventMotion ∗, gpointer)
- void ∗ **SoundOutThread** (void ∗vpData)
- void ∗ **SoundInThread** (void ∗vpData)

Private Member Functions

- void **GetCfg** ()
- void **BuildGUI** ()
- void **ConnectSignals** ()
- bool **ParseServerStr** (char ∗, int ∗, const char ∗)

Private Attributes

- bool **bALSA**
- bool **bFirstTimeout**
- int **iVuTimeout**
- int **iALSACard**
- int **iALSADevice**

- int **iALSASubDevice**
- int **iDeviceBase**
- int **iDCBlock**
- char **cpDevice** [_POSIX_PATH_MAX]
- pthread_t **ptidSoundOut**
- pthread_t **ptidSoundIn** [SUI_MAX_CHANNELS]
- volatile bool **bRun**
- volatile int **iChCount**
- volatile int **iSampleRate**
- volatile int **iOctaveCount**
- volatile long **lDataRefCount**
- volatile long **lSampleCount**
- volatile bool **bpConnected** [SUI_MAX_CHANNELS]
- volatile bool **bpEqEnabled** [SUI_MAX_CHANNELS]
- volatile int **ipSockH** [SUI_MAX_CHANNELS]
- GDT **fpLevelCoeff** [SUI_MAX_CHANNELS]
- clDSPAlloc **ChData** [SUI_MAX_CHANNELS]
- clDSPAlloc **EqCoeffs** [SUI_MAX_CHANNELS]
- guint **guiStatusbarCtxt**
- gfloat **fpInputLevel** [SUI_MAX_CHANNELS]
- gfloat **fpOutputLevel** [SUI_MAX_CHANNELS]
- GtkWidget * **gwWindow**
- GtkWidget * **gwVBox**
- GtkWidget * **gwHBox**
- GtkWidget * **gwStatusbar**
- **clAudio Audio**
- clAudioA2 **AudioA**
- **clCfgFile Cfg**
- clCondition **CondData** [SUI_MAX_CHANNELS]
- clFilter **Filters** [SUI_MAX_CHANNELS]
- **clGtkUtils GtkUtils**
- clMutex **MutexData**
- clMutex **MutexLevel**
- clMutex **MutexChData** [SUI_MAX_CHANNELS]
- clMutex **MutexFilter** [SUI_MAX_CHANNELS]
- clSemaphore **SemStart1**
- clSemaphore **SemStart2**
- **clSockClie SClient**
- **clSoundChGUI** * **SoundChGUI** [SUI_MAX_CHANNELS]

7.87.1 Detailed Description

Sound user interface.

Definition at line 122 of file SoundUI.hh.

7.87.2 Constructor & Destructor Documentation

7.87.2.1 clSoundUI::clSoundUI (int *, char ***)

Definition at line 289 of file SoundUI.cc.

References bFirstTimeout, bpConnected, bRun, cpWindowTxt, lDataRefCount, SUI_MAX_CHANNELS, SUI_VER_MAJ, SUI_VER_MIN, and SUI_VER_PL.

7.87.2.2 clSoundUI::~clSoundUI ()

Definition at line 311 of file SoundUI.cc.

7.87.3 Member Function Documentation

7.87.3.1 void clSoundUI::GetCfg () `[private]`

Definition at line 131 of file SoundUI.cc.

References bALSA, Cfg, cpDevice, clCfgFile::GetInt(), clCfgFile::GetStr(), iALSACard, iALSADevice, iALSASubDevice, iChCount, iDCBlock, iDeviceBase, iSampleRate, iVuTimeout, lSampleCount, clCfgFile::SetFileName(), SUI_CFGFILE, SUI_DEF_CHANNELS, SUI_DEF_DEVICE, SUI_DEF_SAMPLERATE, SUI_SAMPLECOUNT, and SUI_VU_TIMEOUT.

Referenced by Exec().

7.87.3.2 void clSoundUI::BuildGUI () `[private]`

Definition at line 173 of file SoundUI.cc.

References ConnectSignals(), cpWindowTxt, guiStatusbarCtxt, gwHBox, gwStatusbar, gwVBox, gwWindow, iChCount, iOctaveCount, lSampleCount, MutexData, SoundChGUI, and SUI_PADDING.

Referenced by Exec().

7.87.3.3 void clSoundUI::ConnectSignals () `[private]`

Definition at line 222 of file SoundUI.cc.

References gwWindow, iChCount, iOctaveCount, MutexData, SoundChGUI, WrapOnApplyCurveClicked(), WrapOnClickedEvent(), WrapOnDeleteEvent(), WrapOnMotionCurve(), WrapOnToggledEvent(), and WrapOnValueChangedEvent().

Referenced by BuildGUI().

7.87.3.4 bool clSoundUI::ParseServerStr (char *, int *, const char *) [private]

Definition at line 266 of file SoundUI.cc.

References SUI_SERV_MAXLEN.

Referenced by OnClickedEvent().

7.87.3.5 int clSoundUI::Exec ()

Definition at line 316 of file SoundUI.cc.

References bpConnected, bpEqEnabled, BuildGUI(), ChData, CondData, EqCoeffs, Filters, fpLevelCoeff, GDT, GetCfg(), iChCount, iDeviceBase, iOctaveCount, iSampleRate, iVuTimeout, lSampleCount, MutexData, ptidSoundIn, ptidSoundOut, SemStart1, SemStart2, SUI_EQ_MAXOCTS, SUI_FILTER_WINDOW, WrapOnTimeoutEvent(), and WrapSoundOutThread().

Referenced by main().

7.87.3.6 gboolean clSoundUI::OnDeleteEvent (GtkWidget * *gwSender*, GdkEvent * *geEvent*, gpointer *gpData*)

Definition at line 370 of file SoundUI.cc.

References bRun, and MutexData.

Referenced by WrapOnDeleteEvent().

7.87.3.7 void clSoundUI::OnClickedEvent (GtkButton *, gpointer)

Definition at line 381 of file SoundUI.cc.

References bpConnected, clSockClie::Connect(), ipSockH, MutexData, ParseServerStr(), ptidSoundIn, SClient, SoundChGUI, SUI_SERV_MAXLEN, and WrapSoundInThread().

Referenced by WrapOnClickedEvent().

7.87.3.8 gint clSoundUI::OnTimeoutEvent (gpointer)

Definition at line 431 of file SoundUI.cc.

References bFirstTimeout, fpInputLevel, fpOutputLevel, iChCount, MutexData, MutexLevel, and SoundChGUI.

Referenced by WrapOnTimeoutEvent().

7.87.3.9 void clSoundUI::OnToggledEvent (GtkToggleButton *, gpointer)

Definition at line 465 of file SoundUI.cc.

References bpEqEnabled, and MutexFilter.

Referenced by WrapOnToggledEvent().

7.87.3.10 void clSoundUI::OnValueChangedEvent (GtkAdjustment *, gpointer)

Definition at line 479 of file SoundUI.cc.

References EqCoeffs, Filters, fpLevelCoeff, GDT, iDCBlock, iSampleRate, l-SampleCount, MutexData, MutexFilter, and SoundChGUI.

Referenced by WrapOnValueChangedEvent().

7.87.3.11 void clSoundUI::OnApplyCurveClicked (GtkButton *, gpointer)

Definition at line 568 of file SoundUI.cc.

References EqCoeffs, Filters, GDT, iDCBlock, lSampleCount, MutexData, MutexFilter, and SoundChGUI.

Referenced by WrapOnApplyCurveClicked().

7.87.3.12 void clSoundUI::OnMotionCurve (GtkWidget *, GdkEventMotion *, gpointer)

Definition at line 598 of file SoundUI.cc.

References guiStatusbarCtxt, clSoundChGUI::gwCurveEq, gwStatusbar, i-SampleRate, lSampleCount, MutexData, SoundChGUI, SUI_CONV_LEN, and SUI_EQ_RANGE.

Referenced by WrapOnMotionCurve().

7.87.3.13 void * clSoundUI::SoundOutThread (void * *vpData*)

Definition at line 627 of file SoundUI.cc.

References Audio, AUDIO_WRITE, AudioA, bALSA, bRun, ChData, cl-Audio::Close(), CondData, cpDevice, GDT, clAudio::GetError(), iALSACard, iALSADevice, iALSASubDevice, iChCount, iSampleRate, lDataRefCount, lSampleCount, MutexChData, MutexData, clAudio::Open(), ptidSound-Out, clAudio::Reset(), SemStart1, SemStart2, SUI_SND_BITS, SUI_-SND_DATATYPE, SUI_SND_FORMAT, SUI_SND_QUEUESIZE, and cl-Audio::Write().

Referenced by WrapSoundOutThread().

7.87.3.14 void * clSoundUI::SoundInThread (void * *vpData*)

Definition at line 861 of file SoundUI.cc.

References Audio, AUDIO_WRITE, bpConnected, bpEqEnabled, ChData, clAudio::Close(), clSockOp::Close(), CondData, cpDevice, _stSoundStart::d-SampleRate, Filters, fpInputLevel, fpLevelCoeff, fpOutputLevel, GDT, cl-SoundMsg::GetData(), clSockOp::GetErrno(), clAudio::GetOutBufInfo(), cl-SoundMsg::GetStart(), GLOBAL_HEADER_LEN, _stSoundStart::iChannels, i-ChCount, iDeviceBase, ipSockH, lDataRefCount, lSampleCount, MutexCh-Data, MutexData, MutexFilter, MutexLevel, clAudio::Open(), ptidSound-Out, clSockOp::ReadN(), clSockOp::ReadSelect(), clAudio::Reset(), clSock-Op::SetHandle(), clSockOp::SetRecvBufSize(), SOp, SUI_FIRST_TIMEOUT, SUI_IN_TIMEOUT, SUI_SND_DATATYPE, SUI_SND_FORMAT, and cl-Audio::Write().

Referenced by WrapSoundInThread().

7.87.4 Member Data Documentation

7.87.4.1 bool clSoundUI::bALSA `[private]`

Definition at line 124 of file SoundUI.hh.

Referenced by GetCfg(), and SoundOutThread().

7.87.4.2 bool clSoundUI::bFirstTimeout `[private]`

Definition at line 125 of file SoundUI.hh.

Referenced by clSoundUI(), and OnTimeoutEvent().

7.87.4.3 int clSoundUI::iVuTimeout `[private]`

Definition at line 126 of file SoundUI.hh.

Referenced by Exec(), and GetCfg().

7.87.4.4 int clSoundUI::iALSACard `[private]`

Definition at line 128 of file SoundUI.hh.

Referenced by GetCfg(), and SoundOutThread().

7.87.4.5 int clSoundUI::iALSADevice `[private]`

Definition at line 129 of file SoundUI.hh.

Referenced by GetCfg(), and SoundOutThread().

7.87.4.6 int clSoundUI::iALSASubDevice [private]

Definition at line 130 of file SoundUI.hh.

Referenced by GetCfg(), and SoundOutThread().

7.87.4.7 int clSoundUI::iDeviceBase [private]

Definition at line 131 of file SoundUI.hh.

Referenced by Exec(), GetCfg(), and SoundInThread().

7.87.4.8 int clSoundUI::iDCBlock [private]

Definition at line 132 of file SoundUI.hh.

Referenced by GetCfg(), OnApplyCurveClicked(), and OnValueChanged-Event().

7.87.4.9 char clSoundUI::cpDevice[_POSIX_PATH_MAX] [private]

Definition at line 133 of file SoundUI.hh.

Referenced by GetCfg(), SoundInThread(), and SoundOutThread().

7.87.4.10 pthread_t clSoundUI::ptidSoundOut [private]

Definition at line 135 of file SoundUI.hh.

Referenced by Exec(), SoundInThread(), and SoundOutThread().

7.87.4.11 pthread_t clSoundUI::ptidSoundIn[SUI_MAX_-CHANNELS] [private]

Definition at line 136 of file SoundUI.hh.

Referenced by Exec(), and OnClickedEvent().

7.87.4.12 volatile bool clSoundUI::bRun [private]

Definition at line 138 of file SoundUI.hh.

Referenced by clSoundUI(), OnDeleteEvent(), and SoundOutThread().

7.87.4.13 volatile int clSoundUI::iChCount [private]

Definition at line 139 of file SoundUI.hh.

Referenced by BuildGUI(), ConnectSignals(), Exec(), GetCfg(), OnTimeoutEvent(), SoundInThread(), and SoundOutThread().

7.87.4.14 volatile int clSoundUI::iSampleRate [private]

Definition at line 140 of file SoundUI.hh.

Referenced by Exec(), GetCfg(), OnMotionCurve(), OnValueChangedEvent(), and SoundOutThread().

7.87.4.15 volatile int clSoundUI::iOctaveCount [private]

Definition at line 141 of file SoundUI.hh.

Referenced by BuildGUI(), ConnectSignals(), and Exec().

7.87.4.16 volatile long clSoundUI::lDataRefCount [private]

Definition at line 142 of file SoundUI.hh.

Referenced by clSoundUI(), SoundInThread(), and SoundOutThread().

7.87.4.17 volatile long clSoundUI::lSampleCount [private]

Definition at line 143 of file SoundUI.hh.

Referenced by BuildGUI(), Exec(), GetCfg(), OnApplyCurveClicked(), OnMotionCurve(), OnValueChangedEvent(), SoundInThread(), and SoundOutThread().

7.87.4.18 volatile bool clSoundUI::bpConnected[SUI_MAX_CHANNELS] [private]

Definition at line 144 of file SoundUI.hh.

Referenced by clSoundUI(), Exec(), OnClickedEvent(), and SoundInThread().

7.87.4.19 volatile bool clSoundUI::bpEqEnabled[SUI_MAX_CHANNELS] [private]

Definition at line 145 of file SoundUI.hh.

Referenced by Exec(), OnToggledEvent(), and SoundInThread().

7.87.4.20 volatile int clSoundUI::ipSockH[SUI_MAX_CHANNELS] [private]

Definition at line 146 of file SoundUI.hh.

Referenced by OnClickedEvent(), and SoundInThread().

7.87.4.21 GDT clSoundUI::fpLevelCoeff[SUI_MAX_CHANNELS] [private]

Definition at line 147 of file SoundUI.hh.

Referenced by Exec(), OnValueChangedEvent(), and SoundInThread().

7.87.4.22 clDSPAlloc clSoundUI::ChData[SUI_MAX_CHANNELS] [private]

Definition at line 148 of file SoundUI.hh.

Referenced by Exec(), SoundInThread(), and SoundOutThread().

7.87.4.23 clDSPAlloc clSoundUI::EqCoeffs[SUI_MAX_-CHANNELS] [private]

Definition at line 149 of file SoundUI.hh.

Referenced by Exec(), OnApplyCurveClicked(), and OnValueChangedEvent().

7.87.4.24 guint clSoundUI::guiStatusbarCtxt [private]

Definition at line 151 of file SoundUI.hh.

Referenced by BuildGUI(), and OnMotionCurve().

7.87.4.25 gfloat clSoundUI::fpInputLevel[SUI_MAX_CHANNELS] [private]

Definition at line 152 of file SoundUI.hh.

Referenced by OnTimeoutEvent(), and SoundInThread().

7.87.4.26 gfloat clSoundUI::fpOutputLevel[SUI_MAX_-CHANNELS] [private]

Definition at line 153 of file SoundUI.hh.

Referenced by OnTimeoutEvent(), and SoundInThread().

7.87.4.27 GtkWidget* clSoundUI::gwWindow [private]

Definition at line 155 of file SoundUI.hh.

Referenced by BuildGUI(), and ConnectSignals().

7.87.4.28 GtkWidget* clSoundUI::gwVBox [private]

Definition at line 156 of file SoundUI.hh.

Referenced by BuildGUI().

7.87.4.29 GtkWidget* clSoundUI::gwHBox [private]

Definition at line 157 of file SoundUI.hh.

Referenced by BuildGUI().

7.87.4.30 GtkWidget* clSoundUI::gwStatusbar [private]

Definition at line 158 of file SoundUI.hh.

Referenced by BuildGUI(), and OnMotionCurve().

7.87.4.31 clAudio clSoundUI::Audio [private]

Definition at line 161 of file SoundUI.hh.

Referenced by SoundInThread(), and SoundOutThread().

7.87.4.32 clAudioA2 clSoundUI::AudioA [private]

Definition at line 167 of file SoundUI.hh.

Referenced by SoundOutThread().

7.87.4.33 clCfgFile clSoundUI::Cfg [private]

Definition at line 170 of file SoundUI.hh.

Referenced by GetCfg().

7.87.4.34 clCondition clSoundUI::CondData[SUI_MAX_CHANNELS] [private]

Definition at line 171 of file SoundUI.hh.

Referenced by Exec(), SoundInThread(), and SoundOutThread().

7.87.4.35 clFilter clSoundUI::Filters[SUI_MAX_CHANNELS] [private]

Definition at line 172 of file SoundUI.hh.

Referenced by Exec(), OnApplyCurveClicked(), OnValueChangedEvent(), and SoundInThread().

7.87.4.36 clGtkUtils clSoundUI::GtkUtils [private]

Definition at line 173 of file SoundUI.hh.

7.87.4.37 clMutex clSoundUI::MutexData [private]

Definition at line 174 of file SoundUI.hh.

Referenced by BuildGUI(), ConnectSignals(), Exec(), OnApplyCurveClicked(), OnClickedEvent(), OnDeleteEvent(), OnMotionCurve(), OnTimeoutEvent(), OnValueChangedEvent(), SoundInThread(), and SoundOutThread().

7.87.4.38 clMutex clSoundUI::MutexLevel [private]

Definition at line 175 of file SoundUI.hh.

Referenced by OnTimeoutEvent(), and SoundInThread().

7.87.4.39 clMutex clSoundUI::MutexChData[SUI_MAX_-CHANNELS] [private]

Definition at line 176 of file SoundUI.hh.

Referenced by SoundInThread(), and SoundOutThread().

7.87.4.40 clMutex clSoundUI::MutexFilter[SUI_MAX_-CHANNELS] [private]

Definition at line 177 of file SoundUI.hh.

Referenced by OnApplyCurveClicked(), OnToggledEvent(), OnValueChanged-Event(), and SoundInThread().

7.87.4.41 clSemaphore clSoundUI::SemStart1 [private]

Definition at line 178 of file SoundUI.hh.

Referenced by Exec(), and SoundOutThread().

7.87.4.42 clSemaphore clSoundUI::SemStart2 [private]

Definition at line 179 of file SoundUI.hh.

Referenced by Exec(), and SoundOutThread().

7.87.4.43 clSockClie clSoundUI::SClient [private]

Definition at line 180 of file SoundUI.hh.

Referenced by OnClickedEvent().

7.87.4.44 clSoundChGUI* clSoundUI::SoundChGUI[SUI_MAX_-CHANNELS] [private]

Definition at line 181 of file SoundUI.hh.

Referenced by BuildGUI(), ConnectSignals(), OnApplyCurveClicked(), OnClickedEvent(), OnMotionCurve(), OnTimeoutEvent(), and OnValueChangedEvent().

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

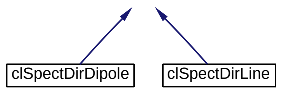
- **SoundUI.hh**
- **SoundUI.cc**

7.88 clSpectDir Class Reference

Direction calculation from spectrum.

`#include <SpectDir.hh>`

Inheritance diagram for clSpectDir:



Public Member Functions

- **clSpectDir** ()
- ~**clSpectDir** ()
- void **SetSensorSpacing** (GDT)

    Set sensor spacing.

- void **SetSoundSpeed** (GDT)

    Set speed of sound.

- GDT **GetArrayFrequency** ()

    Get array frequency.

- GDT **GetDirection** (GDT, GDT)

    Get direction from frequency and phase difference.

- void **SetDirectionCount** (long)

    Set number of directions.

- void **SetDirection** (GDT, GDT, GDT)

    Set direction from frequency, power/magnitude and phase diff.

- GDT * **GetDirections** (GDT *)

    Get vector containing results of subsequent **SetDirection()** (p. 517) *'s.*

- void **ResetDirections** ()

    Reset directions.

Public Attributes

- clDSPOp **DSP**

Private Attributes

- long **lDirCount**
- GDT **fPI**
- GDT **fHalfPI**
- GDT **fSensorSpacing**
- GDT **fArrayFreq**
- GDT * **fpDirections**
- clAlloc **DirBuf**

7.88.1 Detailed Description

Direction calculation from spectrum.

This is 100% my (Jussi Laako) own algorithm. At least I haven't seen this anywhere before and I'm not aware if such exists anywhere else.

Definition at line 53 of file SpectDir.hh.

7.88.2 Constructor & Destructor Documentation

7.88.2.1 clSpectDir::clSpectDir ()

Definition at line 30 of file SpectDir.cc.

References fHalfPI, fpDirections, fPI, and lDirCount.

7.88.2.2 clSpectDir::~clSpectDir ()

Definition at line 39 of file SpectDir.cc.

7.88.3 Member Function Documentation

7.88.3.1 void clSpectDir::SetSensorSpacing (GDT)

Set sensor spacing.

Note:
Call this before setting speed of sound.

Parameters:
fSpacing Sensor spacing (m)

Definition at line 44 of file SpectDir.cc.

References fSensorSpacing.

Referenced by clSpectDirDipole::clSpectDirDipole(), and clSpectDirLine::clSpectDirLine().

7.88.3.2 void clSpectDir::SetSoundSpeed (GDT)

Set speed of sound.

Parameters:
fSoundSpeed Speed of sound (m/s)

Definition at line 50 of file SpectDir.cc.

References fArrayFreq, and fSensorSpacing.

Referenced by clSpectDirDipole::clSpectDirDipole(), and clSpectDirLine::clSpectDirLine().

7.88.3.3 GDT clSpectDir::GetArrayFrequency () `[inline]`

Get array frequency.

Get maximum frequency that can be handled by the array without false directions. It is $\frac{\lambda}{2}$.

Returns:
Array frequency

Definition at line 88 of file SpectDir.hh.

References fArrayFreq, and GDT.

Referenced by clSpectDirDipole::clSpectDirDipole(), and clSpectDirLine::clSpectDirLine().

7.88.3.4 GDT clSpectDir::GetDirection (GDT, GDT)

Get direction from frequency and phase difference.

$$\Delta f = \frac{f_a}{f}$$

$$\theta = \Delta f \frac{\Delta \varphi}{2}$$

Parameters:
fFrequency Frequency (Hz)
fDPhase Phase difference (rad)

Returns:
Direction (rad)

Definition at line 56 of file SpectDir.cc.

References fArrayFreq, and GDT.

Referenced by SetDirection().

7.88.3.5 void clSpectDir::SetDirectionCount (long)

Set number of directions.

180 / n = resolution.

Parameters:

lCount Number of directions (sectors)

Definition at line 67 of file SpectDir.cc.

References DirBuf, DSP, fpDirections, GDT, and lDirCount.

Referenced by clSpectDirDipole::clSpectDirDipole(), and clSpectDirLine::clSpectDirLine().

7.88.3.6 void clSpectDir::SetDirection (GDT, GDT, GDT)

Set direction from frequency, power/magnitude and phase diff.

Note:

Results are integrated over subsequent calls.

Parameters:

fFrequency Frequency (Hz)

fLevel Level

fDPhase Phase difference

Definition at line 75 of file SpectDir.cc.

References DSP, fpDirections, fPI, GDT, GetDirection(), and lDirCount.

Referenced by clSpectDirLine::Calculate(), and clSpectDirDipole::Calculate().

7.88.3.7 GDT * clSpectDir::GetDirections (GDT *)

Get vector containing results of subsequent **SetDirection()**(p. 517)'s.

Note:

Destination pointer can be NULL, destination copy is normalized to [0:1], returned pointer is pointer to raw results.

Parameters:

fpDest Directions

Returns:

Pointer to directions

Definition at line 89 of file SpectDir.cc.

References DSP, fpDirections, GDT, and lDirCount.

Referenced by clSpectDirLine::GetResults(), and clSpectDirDipole::GetResults().

7.88.3.8 void clSpectDir::ResetDirections ()

Reset directions.

Definition at line 96 of file SpectDir.cc.

References DSP, fpDirections, and lDirCount.

Referenced by clSpectDirLine::ResetResults(), and clSpectDirDipole::ResetResults().

7.88.4 Member Data Documentation

7.88.4.1 long clSpectDir::lDirCount `[private]`

Definition at line 55 of file SpectDir.hh.

Referenced by clSpectDir(), GetDirections(), ResetDirections(), SetDirection(), and SetDirectionCount().

7.88.4.2 GDT clSpectDir::fPI `[private]`

Definition at line 56 of file SpectDir.hh.

Referenced by clSpectDir(), and SetDirection().

7.88.4.3 GDT clSpectDir::fHalfPI `[private]`

Definition at line 57 of file SpectDir.hh.

Referenced by clSpectDir().

7.88.4.4 GDT clSpectDir::fSensorSpacing `[private]`

Definition at line 58 of file SpectDir.hh.

Referenced by SetSensorSpacing(), and SetSoundSpeed().

7.88.4.5 GDT clSpectDir::fArrayFreq `[private]`

Definition at line 59 of file SpectDir.hh.

Referenced by GetArrayFrequency(), GetDirection(), and SetSoundSpeed().

7.88.4.6 GDT* clSpectDir::fpDirections `[private]`

Definition at line 60 of file SpectDir.hh.

Referenced by clSpectDir(), GetDirections(), ResetDirections(), SetDirection(), and SetDirectionCount().

7.88.4.7 clAlloc clSpectDir::DirBuf [private]

Definition at line 61 of file SpectDir.hh.

Referenced by SetDirectionCount().

7.88.4.8 clDSPOp clSpectDir::DSP

Definition at line 63 of file SpectDir.hh.

Referenced by clSpectDirLine::Calculate(), clSpectDirDipole::Calculate(), clSpectDirDipole::clSpectDirDipole(), clSpectDirLine::clSpectDirLine(), GetDirections(), clSpectDirLine::GetResults(), clSpectDirDipole::GetResults(), clSpectDirLine::PutData(), clSpectDirDipole::PutData(), ResetDirections(), SetDirection(), and SetDirectionCount().

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

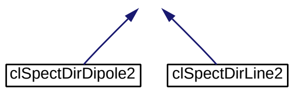
- **SpectDir.hh**
- **SpectDir.cc**

7.89 clSpectDir2 Class Reference

Direction calculation from spectrum.

`#include <SpectDir2.hh>`

Inheritance diagram for clSpectDir2:



Public Member Functions

- **clSpectDir2** ()
- ~**clSpectDir2** ()
- void **SetSensorSpacing** (GDT)
- void **SetSoundSpeed** (GDT)
- GDT **GetArrayFrequency** ()
- GDT **GetDirection** (GDT, GDT)

Public Attributes

- clDSPOp **DSP**

Private Attributes

- long **lDirCount**
- GDT **fPI**
- GDT **fSensorSpacing**
- GDT **fArrayFreq**

7.89.1 Detailed Description

Direction calculation from spectrum.

See **clSpectDir**(p. 514) for details.

Definition at line 47 of file SpectDir2.hh.

7.89.2 Constructor & Destructor Documentation

7.89.2.1 clSpectDir2::clSpectDir2 ()

Definition at line 30 of file SpectDir2.cc.

References fPI.

7.89.2.2 clSpectDir2::∼clSpectDir2 ()

Definition at line 36 of file SpectDir2.cc.

7.89.3 Member Function Documentation

7.89.3.1 void clSpectDir2::SetSensorSpacing (GDT)

Definition at line 41 of file SpectDir2.cc.

References fSensorSpacing.

Referenced by clSpectDirDipole2::clSpectDirDipole2(), and clSpectDirLine2::clSpectDirLine2().

7.89.3.2 void clSpectDir2::SetSoundSpeed (GDT)

Definition at line 47 of file SpectDir2.cc.

References fArrayFreq, and fSensorSpacing.

Referenced by clSpectDirDipole2::clSpectDirDipole2(), and clSpectDirLine2::clSpectDirLine2().

7.89.3.3 GDT clSpectDir2::GetArrayFrequency () `[inline]`

Definition at line 65 of file SpectDir2.hh.

References fArrayFreq, and GDT.

Referenced by clSpectDirDipole2::clSpectDirDipole2(), and clSpectDirLine2::clSpectDirLine2().

7.89.3.4 GDT clSpectDir2::GetDirection (GDT, GDT)

Definition at line 53 of file SpectDir2.cc.

References fArrayFreq, and GDT.

Referenced by clSpectDirLine2::Calculate(), and clSpectDirDipole2::Calculate().

7.89.4 Member Data Documentation

7.89.4.1 long clSpectDir2::lDirCount `[private]`

Definition at line 49 of file SpectDir2.hh.

7.89.4.2 GDT clSpectDir2::fPI [private]

Definition at line 50 of file SpectDir2.hh.

Referenced by clSpectDir2().

7.89.4.3 GDT clSpectDir2::fSensorSpacing [private]

Definition at line 51 of file SpectDir2.hh.

Referenced by SetSensorSpacing(), and SetSoundSpeed().

7.89.4.4 GDT clSpectDir2::fArrayFreq [private]

Definition at line 52 of file SpectDir2.hh.

Referenced by GetArrayFrequency(), GetDirection(), and SetSoundSpeed().

7.89.4.5 clDSPOp clSpectDir2::DSP

Definition at line 54 of file SpectDir2.hh.

Referenced by clSpectDirLine2::Calculate(), clSpectDirDipole2::Calculate(), clSpectDirDipole2::clSpectDirDipole2(), clSpectDirLine2::clSpectDirLine2(), clSpectDirLine2::GetResults(), clSpectDirDipole2::GetResults(), clSpectDirLine2::PutData(), and clSpectDirDipole2::PutData().

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

- **SpectDir2.hh**
- **SpectDir2.cc**

7.90 _stSpectDir2RN Struct Reference

`#include <SpectDir2.hh>`

Public Attributes

- int **iType**
- GDT **fAlpha**
- long **lMeanLength**
- long **lGapLength**

7.90.1 Member Data Documentation

7.90.1.1 int _stSpectDir2RN::iType

Definition at line 35 of file SpectDir2.hh.

Referenced by clSpectDirLine2::Calculate(), clSpectDirDipole2::Calculate(), and clDirection3::ProcessLoop().

7.90.1.2 GDT _stSpectDir2RN::fAlpha

Definition at line 36 of file SpectDir2.hh.

Referenced by clSpectDirLine2::Calculate(), clSpectDirDipole2::Calculate(), and clDirection3::ProcessLoop().

7.90.1.3 long _stSpectDir2RN::lMeanLength

Definition at line 37 of file SpectDir2.hh.

Referenced by clSpectDirLine2::Calculate(), clSpectDirDipole2::Calculate(), and clDirection3::ProcessLoop().

7.90.1.4 long _stSpectDir2RN::lGapLength

Definition at line 38 of file SpectDir2.hh.

Referenced by clSpectDirLine2::Calculate(), clSpectDirDipole2::Calculate(), and clDirection3::ProcessLoop().

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

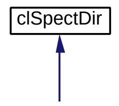
- **SpectDir2.hh**

7.91 clSpectDirDipole Class Reference

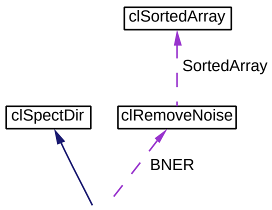
Spectrum based direction finding for a dipole array.

`#include <SpectDirDipole.hh>`

Inheritance diagram for clSpectDirDipole:



Collaboration diagram for clSpectDirDipole:



Public Member Functions

- **clSpectDirDipole** (GDT, GDT, double, long, int, long, long, GDT, bool)

  Constructor.

- **~clSpectDirDipole** ()
- void **PutData** (const GDT *, long, long, long)

  Put data into input data FIFO.

- bool **GetResults** (GDT *, GDT, int, **stpSpectDirRN**)

  Get results.

- void **ResetResults** ()

  Reset results.

- GDT **GetIntegrationTime** ()

  Get integration time (seconds).

Private Member Functions

- void **Calculate** (int, **stpSpectDirRN**)

Private Attributes

- bool **bDebug**
- long **lDirectionCount**
- long **lFilterSize**
- long **lFFTSize**
- long **lDecimation**
- long **lSpectSize**
- long **lMinBin**
- long **lMaxBin**
- long **lNewData**
- long **lOldData**
- float **fOverlap**
- GDT **fFreqRes**
- GDT **fIntTime**
- GDT * **fpWinFunc**
- GDT * **fpExtBuf**
- GDT * **fpProcBuf** [2]
- GDT * **fpPrevBuf** [2]
- GDT * **fpRNBuf** [2]
- GCDT * **spSpect** [2]
- clDSPAlloc **WinFuncBuf**
- clDSPAlloc **ExtBuf**
- clDSPAlloc **ProcBuf** [2]
- clDSPAlloc **PrevBuf** [2]
- clDSPAlloc **RNBuf** [2]
- clDSPAlloc **SpectBuf** [2]
- clRecDecimator **Decimator** [2]
- **clRemoveNoise BNER**

7.91.1 Detailed Description

Spectrum based direction finding for a dipole array.

This is 100% my (Jussi Laako) own algorithm. At least I haven't seen this anywhere before and I'm not aware if such exists anywhere else.

Definition at line 66 of file SpectDirDipole.hh.

7.91.2 Constructor & Destructor Documentation

7.91.2.1 clSpectDirDipole::clSpectDirDipole (GDT, GDT, double, long, int, long, long, GDT, bool)

Constructor.

Parameters:

fSensorSpacing Sensor distance (m)

fSoundSpeed Speed of sound (m/s)

dSampleRate Samplerate

lFiltSize Filter size (points)

iFilterType Type of decimation filter

lWindowSize Window size (points)

lDirCount Number of sectors

fIntTimeReq Integration time request (s)

bEnableDebug Enable debug?

Definition at line 104 of file SpectDirDipole.cc.

References bDebug, Decimator, clSpectDir::DSP, fFreqRes, fIntTime, fOverlap, fpPrevBuf, fpProcBuf, fpRNBuf, fpWinFunc, GCDT, GDT, clSpectDir::GetArrayFrequency(), lDecimation, lDirectionCount, lFFTSize, lFilterSize, lMaxBin, lNewData, lOldData, lSpectSize, PrevBuf, ProcBuf, RNBuf, clSpectDir::SetDirectionCount(), clSpectDir::SetSensorSpacing(), clSpectDir::SetSoundSpeed(), SpectBuf, spSpect, and WinFuncBuf.

7.91.2.2 clSpectDirDipole::~clSpectDirDipole ()

Definition at line 186 of file SpectDirDipole.cc.

7.91.3 Member Function Documentation

7.91.3.1 void clSpectDirDipole::Calculate (int, stpSpectDirRN) `[private]`

Definition at line 30 of file SpectDirDipole.cc.

References BNER, clRemoveNoise::Diff(), clSpectDir::DSP, _stSpectDirRN::fAlpha, fFreqRes, fpProcBuf, fpRNBuf, fpWinFunc, GCDT, GDT, clRemoveNoise::InvDiff(), _stSpectDirRN::iType, lFFTSize, _stSpectDirRN::lGapLength, lMaxBin, _stSpectDirRN::lMeanLength, lMinBin, lSpectSize, clRemoveNoise::OTA(), SDD_BNER_DIFF, SDD_BNER_IDIFF, SDD_BNER_NONE, SDD_BNER_OTA, SDD_BNER_TPSW, SDD_SCALE_LIN, SDD_SCALE_LOG, clSpectDir::SetDirection(), spSpect, and clRemoveNoise::TPSW().

Referenced by GetResults().

7.91.3.2 void clSpectDirDipole::PutData (const GDT *, long, long, long)

Put data into input data FIFO.

Parameters:

fpInputData Input data

lDataCount Number of samples (total)

lStartCh Starting channel index

lChCount Number of channels (total)

Definition at line 191 of file SpectDirDipole.cc.

References Decimator, clSpectDir::DSP, ExtBuf, fpExtBuf, and GDT.

Referenced by clDirection2::ProcessLoop().

7.91.3.3 bool clSpectDirDipole::GetResults (GDT *, GDT, int, stpSpectDirRN)

Get results.

$$V_f(x) = \sqrt{\left(\Re F(x)_L{}^2 + \Im F(x)_L{}^2\right) + \left(\Re F(x)_R{}^2 + \Im F(x)_R{}^2\right)}$$

$$z_c(x) = F(x)_L F(x)_R^*$$

$$\Delta\varphi_f(x) = \arctan\left(\frac{\Im z_c(x)}{\Re z_c(x)}\right)$$

Parameters:

fpResults Results vector

fLowFreqLimit Lower frequency limit (Hz)

iScaling Scaling type

spRemoveNoise Noise removal parameters

Returns:

Results available?

Definition at line 209 of file SpectDirDipole.cc.

References Calculate(), Decimator, clSpectDir::DSP, fFreqRes, fpPrevBuf, fp-ProcBuf, GDT, clSpectDir::GetDirections(), lFFTSize, lMaxBin, lMinBin, l-NewData, and lOldData.

Referenced by clDirection2::ProcessLoop().

7.91.3.4 void clSpectDirDipole::ResetResults ()

Reset results.

Definition at line 243 of file SpectDirDipole.cc.

References clSpectDir::ResetDirections().

Referenced by clDirection2::ProcessLoop().

7.91.3.5 GDT clSpectDirDipole::GetIntegrationTime () `[inline]`

Get integration time (seconds).

Returns:
Integration time (s)

Definition at line 145 of file SpectDirDipole.hh.

References fIntTime, and GDT.

Referenced by clDirection2::ProcessLoop().

7.91.4 Member Data Documentation

7.91.4.1 bool clSpectDirDipole::bDebug `[private]`

Definition at line 68 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.2 long clSpectDirDipole::lDirectionCount `[private]`

Definition at line 69 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.3 long clSpectDirDipole::lFilterSize `[private]`

Definition at line 70 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.4 long clSpectDirDipole::lFFTSize `[private]`

Definition at line 71 of file SpectDirDipole.hh.

Referenced by Calculate(), clSpectDirDipole(), and GetResults().

7.91.4.5 long clSpectDirDipole::lDecimation [private]

Definition at line 72 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.6 long clSpectDirDipole::lSpectSize [private]

Definition at line 73 of file SpectDirDipole.hh.

Referenced by Calculate(), and clSpectDirDipole().

7.91.4.7 long clSpectDirDipole::lMinBin [private]

Definition at line 74 of file SpectDirDipole.hh.

Referenced by Calculate(), and GetResults().

7.91.4.8 long clSpectDirDipole::lMaxBin [private]

Definition at line 75 of file SpectDirDipole.hh.

Referenced by Calculate(), clSpectDirDipole(), and GetResults().

7.91.4.9 long clSpectDirDipole::lNewData [private]

Definition at line 76 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole(), and GetResults().

7.91.4.10 long clSpectDirDipole::lOldData [private]

Definition at line 77 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole(), and GetResults().

7.91.4.11 float clSpectDirDipole::fOverlap [private]

Definition at line 78 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.12 GDT clSpectDirDipole::fFreqRes [private]

Definition at line 79 of file SpectDirDipole.hh.

Referenced by Calculate(), clSpectDirDipole(), and GetResults().

7.91.4.13 GDT clSpectDirDipole::fIntTime [private]

Definition at line 80 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole(), and GetIntegrationTime().

7.91.4.14 GDT* clSpectDirDipole::fpWinFunc [private]

Definition at line 81 of file SpectDirDipole.hh.

Referenced by Calculate(), and clSpectDirDipole().

7.91.4.15 GDT* clSpectDirDipole::fpExtBuf [private]

Definition at line 82 of file SpectDirDipole.hh.

Referenced by PutData().

7.91.4.16 GDT* clSpectDirDipole::fpProcBuf[2] [private]

Definition at line 83 of file SpectDirDipole.hh.

Referenced by Calculate(), clSpectDirDipole(), and GetResults().

7.91.4.17 GDT* clSpectDirDipole::fpPrevBuf[2] [private]

Definition at line 84 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole(), and GetResults().

7.91.4.18 GDT* clSpectDirDipole::fpRNBuf[2] [private]

Definition at line 85 of file SpectDirDipole.hh.

Referenced by Calculate(), and clSpectDirDipole().

7.91.4.19 GCDT* clSpectDirDipole::spSpect[2] [private]

Definition at line 86 of file SpectDirDipole.hh.

Referenced by Calculate(), and clSpectDirDipole().

7.91.4.20 clDSPAlloc clSpectDirDipole::WinFuncBuf [private]

Definition at line 87 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.21 clDSPAlloc clSpectDirDipole::ExtBuf [private]

Definition at line 88 of file SpectDirDipole.hh.

Referenced by PutData().

7.91.4.22 clDSPAlloc clSpectDirDipole::ProcBuf[2] [private]

Definition at line 89 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.23 clDSPAlloc clSpectDirDipole::PrevBuf[2] [private]

Definition at line 90 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.24 clDSPAlloc clSpectDirDipole::RNBuf[2] [private]

Definition at line 91 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.25 clDSPAlloc clSpectDirDipole::SpectBuf[2] [private]

Definition at line 92 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole().

7.91.4.26 clRecDecimator clSpectDirDipole::Decimator[2] [private]

Definition at line 93 of file SpectDirDipole.hh.

Referenced by clSpectDirDipole(), GetResults(), and PutData().

7.91.4.27 clRemoveNoise clSpectDirDipole::BNER [private]

Definition at line 94 of file SpectDirDipole.hh.

Referenced by Calculate().

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

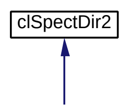
- **SpectDirDipole.hh**
- **SpectDirDipole.cc**

7.92 clSpectDirDipole2 Class Reference

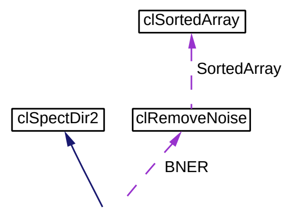
Spectrum based direction finding for dipole array.

`#include <SpectDirDipole2.hh>`

Inheritance diagram for clSpectDirDipole2:



Collaboration diagram for clSpectDirDipole2:



Public Member Functions

- **clSpectDirDipole2** (GDT, GDT, double, long, int, long, GDT, bool)

    Constructor.

- ~**clSpectDirDipole2** ()
- void **PutData** (const GDT ∗, long, long, long)

    Put data into input FIFO.

- bool **GetResults** (GDT ∗, GDT ∗, GDT, int, **stpSpectDir2RN**)

    Get copy of results.

- GDT ∗ **GetLevels** ()

    Get pointer to level results.

- GDT ∗ **GetDirections** ()

    Get pointer to direction results.

- GDT **GetIntegrationTime** ()

    Get integration time (seconds).

- GDT **GetFreqResolution** ()

  Get frequency resolution (Hz).

- long **GetMinBin** ()

  Get minimum used spectrum bin.

- long **GetMaxBin** ()

  Get maximum used spectrum bin.

- long **GetResultCount** ()

  Get result count.

Private Member Functions

- void **Calculate** (int, **stpSpectDir2RN**)

Private Attributes

- bool **bDebug**
- long **lFilterSize**
- long **lFFTSize**
- long **lDecimation**
- long **lSpectSize**
- long **lMinBin**
- long **lMaxBin**
- long **lNewData**
- long **lOldData**
- float **fOverlap**
- GDT **fFreqRes**
- GDT **fIntTime**
- GDT * **fpWinFunc**
- GDT * **fpLevRes**
- GDT * **fpDirRes**
- GDT * **fpExtBuf**
- GDT * **fpProcBuf** [2]
- GDT * **fpPrevBuf** [2]
- GDT * **fpRNBuf** [2]
- GCDT * **spSpect** [2]
- clDSPAlloc **WinFuncBuf**
- clDSPAlloc **LevResBuf**
- clDSPAlloc **DirResBuf**
- clDSPAlloc **ExtBuf**
- clDSPAlloc **ProcBuf** [2]
- clDSPAlloc **PrevBuf** [2]
- clDSPAlloc **RNBuf** [2]

- clDSPAlloc **SpectBuf** [2]
- clRecDecimator **Decimator** [2]
- **clRemoveNoise BNER**

7.92.1 Detailed Description

Spectrum based direction finding for dipole array.

This is used by **clDirection3**(p. 185).

Note:
See **clSpectDirDipole**(p. 524) for details!

Definition at line 61 of file SpectDirDipole2.hh.

7.92.2 Constructor & Destructor Documentation

7.92.2.1 clSpectDirDipole2::clSpectDirDipole2 (GDT, GDT, double, long, int, long, GDT, bool)

Constructor.

Parameters:

fSensorSpacing Sensor spacing (m)

fSoundSpeed Speed of sound (m/s)

dSampleRate Samplerate

lFiltSize Size of filter (points)

iFilterType Type of decimation filter

lWindowSize Size of window (points)

fIntTimeReq Integration time request (s)

bEnableDebug Enable debug?

Definition at line 105 of file SpectDirDipole2.cc.

References bDebug, Decimator, DirResBuf, clSpectDir2::DSP, fFreqRes, fIntTime, fOverlap, fpDirRes, fpLevRes, fpPrevBuf, fpProcBuf, fpRNBuf, fpWinFunc, GCDT, GDT, clSpectDir2::GetArrayFrequency(), lDecimation, LevResBuf, lFFTSize, lFilterSize, lMaxBin, lNewData, lOldData, lSpectSize, PrevBuf, ProcBuf, RNBuf, clSpectDir2::SetSensorSpacing(), clSpectDir2::SetSoundSpeed(), SpectBuf, spSpect, and WinFuncBuf.

7.92.2.2 clSpectDirDipole2::~clSpectDirDipole2 ()

Definition at line 186 of file SpectDirDipole2.cc.

7.92.3 Member Function Documentation

7.92.3.1 void clSpectDirDipole2::Calculate (int, stpSpectDir2RN) [private]

Definition at line 30 of file SpectDirDipole2.cc.

References BNER, clRemoveNoise::Diff(), clSpectDir2::DSP, _stSpectDir2RN::fAlpha, fFreqRes, fpDirRes, fpLevRes, fpProcBuf, fpRNBuf, fpWinFunc, GCDT, GDT, clSpectDir2::GetDirection(), clRemoveNoise::InvDiff(), _stSpectDir2RN::iType, lFFTSize, _stSpectDir2RN::lGapLength, lMaxBin, _stSpectDir2RN::lMeanLength, lMinBin, lSpectSize, clRemoveNoise::OTA(), SDD2_BNER_DIFF, SDD2_BNER_IDIFF, SDD2_BNER_NONE, SDD2_BNER_OTA, SDD2_BNER_TPSW, SDD2_SCALE_LIN, SDD2_SCALE_LOG, spSpect, and clRemoveNoise::TPSW().

Referenced by GetResults().

7.92.3.2 void clSpectDirDipole2::PutData (const GDT *, long, long, long)

Put data into input FIFO.

Parameters:

fpInputData Input data vector

lSampleCount Number of sample in input vector (total)

lStartCh Starting channel index

lChCount Channel count, total

Definition at line 191 of file SpectDirDipole2.cc.

References Decimator, clSpectDir2::DSP, ExtBuf, fpExtBuf, and GDT.

Referenced by clDirection3::ProcessLoop().

7.92.3.3 bool clSpectDirDipole2::GetResults (GDT *, GDT *, GDT, int, stpSpectDir2RN)

Get copy of results.

Parameters:

fpLevelResults Levels vector (can be NULL)

fpDirResults Directions vector (can be NULL)

fLowFreqLimit Low frequency limit (Hz)

iScaling Scaling type

spRemoveNoise Noise removal parameters

Returns:

Results available?

Definition at line 208 of file SpectDirDipole2.cc.

References Calculate(), Decimator, clSpectDir2::DSP, fFreqRes, fpDirRes, fpLevRes, fpPrevBuf, fpProcBuf, GDT, lFFTSize, lMaxBin, lMinBin, lNewData, lOldData, and lSpectSize.

Referenced by clDirection3::ProcessLoop().

7.92.3.4 GDT* clSpectDirDipole2::GetLevels () `[inline]`

Get pointer to level results.

Returns:
Pointer to level results

Definition at line 134 of file SpectDirDipole2.hh.

References fpLevRes, and GDT.

Referenced by clDirection3::ProcessLoop().

7.92.3.5 GDT* clSpectDirDipole2::GetDirections () `[inline]`

Get pointer to direction results.

Returns:
Pointer to direction results

Definition at line 140 of file SpectDirDipole2.hh.

References fpDirRes, and GDT.

Referenced by clDirection3::ProcessLoop().

7.92.3.6 GDT clSpectDirDipole2::GetIntegrationTime () `[inline]`

Get integration time (seconds).

Returns:
Integration time (s)

Definition at line 146 of file SpectDirDipole2.hh.

References fIntTime, and GDT.

Referenced by clDirection3::ProcessLoop().

7.92.3.7 GDT clSpectDirDipole2::GetFreqResolution () `[inline]`

Get frequency resolution (Hz).

Returns:
Frequency resolution (Hz)

Definition at line 152 of file SpectDirDipole2.hh.

References fFreqRes, and GDT.

Referenced by clDirection3::ProcessLoop().

7.92.3.8 long clSpectDirDipole2::GetMinBin () [inline]

Get minimum used spectrum bin.

Returns:
Index to lowest used bin

Definition at line 158 of file SpectDirDipole2.hh.

References lMinBin.

Referenced by clDirection3::ProcessLoop().

7.92.3.9 long clSpectDirDipole2::GetMaxBin () [inline]

Get maximum used spectrum bin.

Returns:
Index to highest used bin

Definition at line 164 of file SpectDirDipole2.hh.

References lMaxBin.

Referenced by clDirection3::ProcessLoop().

7.92.3.10 long clSpectDirDipole2::GetResultCount () [inline]

Get result count.

Returns:
Result vector length

Definition at line 170 of file SpectDirDipole2.hh.

References lSpectSize.

Referenced by clDirection3::ProcessLoop().

7.92.4 Member Data Documentation

7.92.4.1 bool clSpectDirDipole2::bDebug [private]

Definition at line 63 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.2 long clSpectDirDipole2::lFilterSize [private]

Definition at line 64 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.3 long clSpectDirDipole2::lFFTSize [private]

Definition at line 65 of file SpectDirDipole2.hh.

Referenced by Calculate(), clSpectDirDipole2(), and GetResults().

7.92.4.4 long clSpectDirDipole2::lDecimation [private]

Definition at line 66 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.5 long clSpectDirDipole2::lSpectSize [private]

Definition at line 67 of file SpectDirDipole2.hh.

Referenced by Calculate(), clSpectDirDipole2(), GetResultCount(), and GetResults().

7.92.4.6 long clSpectDirDipole2::lMinBin [private]

Definition at line 68 of file SpectDirDipole2.hh.

Referenced by Calculate(), GetMinBin(), and GetResults().

7.92.4.7 long clSpectDirDipole2::lMaxBin [private]

Definition at line 69 of file SpectDirDipole2.hh.

Referenced by Calculate(), clSpectDirDipole2(), GetMaxBin(), and GetResults().

7.92.4.8 long clSpectDirDipole2::lNewData [private]

Definition at line 70 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2(), and GetResults().

7.92.4.9 long clSpectDirDipole2::lOldData [private]

Definition at line 71 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2(), and GetResults().

7.92.4.10 float clSpectDirDipole2::fOverlap [private]

Definition at line 72 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.11 GDT clSpectDirDipole2::fFreqRes [private]

Definition at line 73 of file SpectDirDipole2.hh.

Referenced by Calculate(), clSpectDirDipole2(), GetFreqResolution(), and GetResults().

7.92.4.12 GDT clSpectDirDipole2::fIntTime [private]

Definition at line 74 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2(), and GetIntegrationTime().

7.92.4.13 GDT* clSpectDirDipole2::fpWinFunc [private]

Definition at line 75 of file SpectDirDipole2.hh.

Referenced by Calculate(), and clSpectDirDipole2().

7.92.4.14 GDT* clSpectDirDipole2::fpLevRes [private]

Definition at line 76 of file SpectDirDipole2.hh.

Referenced by Calculate(), clSpectDirDipole2(), GetLevels(), and GetResults().

7.92.4.15 GDT* clSpectDirDipole2::fpDirRes [private]

Definition at line 77 of file SpectDirDipole2.hh.

Referenced by Calculate(), clSpectDirDipole2(), GetDirections(), and GetResults().

7.92.4.16 GDT* clSpectDirDipole2::fpExtBuf [private]

Definition at line 78 of file SpectDirDipole2.hh.

Referenced by PutData().

7.92.4.17 GDT* clSpectDirDipole2::fpProcBuf[2] [private]

Definition at line 79 of file SpectDirDipole2.hh.

Referenced by Calculate(), clSpectDirDipole2(), and GetResults().

7.92.4.18 GDT* clSpectDirDipole2::fpPrevBuf[2] [private]

Definition at line 80 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2(), and GetResults().

7.92.4.19 GDT* clSpectDirDipole2::fpRNBuf[2] [private]

Definition at line 81 of file SpectDirDipole2.hh.

Referenced by Calculate(), and clSpectDirDipole2().

7.92.4.20 GCDT* clSpectDirDipole2::spSpect[2] [private]

Definition at line 82 of file SpectDirDipole2.hh.

Referenced by Calculate(), and clSpectDirDipole2().

7.92.4.21 clDSPAlloc clSpectDirDipole2::WinFuncBuf [private]

Definition at line 83 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.22 clDSPAlloc clSpectDirDipole2::LevResBuf [private]

Definition at line 84 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.23 clDSPAlloc clSpectDirDipole2::DirResBuf [private]

Definition at line 85 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.24 clDSPAlloc clSpectDirDipole2::ExtBuf [private]

Definition at line 86 of file SpectDirDipole2.hh.

Referenced by PutData().

7.92.4.25 clDSPAlloc clSpectDirDipole2::ProcBuf[2] [private]

Definition at line 87 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.26 clDSPAlloc clSpectDirDipole2::PrevBuf[2] [private]

Definition at line 88 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.27 clDSPAlloc clSpectDirDipole2::RNBuf[2] [private]

Definition at line 89 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.28 clDSPAlloc clSpectDirDipole2::SpectBuf[2] [private]

Definition at line 90 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2().

7.92.4.29 clRecDecimator clSpectDirDipole2::Decimator[2] [private]

Definition at line 91 of file SpectDirDipole2.hh.

Referenced by clSpectDirDipole2(), GetResults(), and PutData().

7.92.4.30 clRemoveNoise clSpectDirDipole2::BNER [private]

Definition at line 92 of file SpectDirDipole2.hh.

Referenced by Calculate().

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

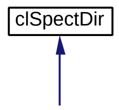
- **SpectDirDipole2.hh**
- **SpectDirDipole2.cc**

7.93 clSpectDirLine Class Reference

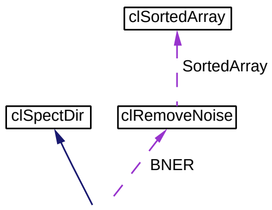
Spectrum based direction finding for a line array.

`#include <SpectDirLine.hh>`

Inheritance diagram for clSpectDirLine:



Collaboration diagram for clSpectDirLine:



Public Member Functions

- **clSpectDirLine** (long, GDT, GDT, double, long, int, long, long, GDT, bool)

    Constructor.

- **~clSpectDirLine** ()
- void **PutData** (const GDT *, long, long, long)

    Put data into input data FIFO.

- bool **GetResults** (GDT *, GDT, int, **stpSpectDirRN**)

    Get results.

- void **ResetResults** ()

    Reset results.

- GDT **GetIntegrationTime** ()

    Get integration time (seconds).

Private Member Functions

- void **Calculate** (int, **stpSpectDirRN**)

Private Attributes

- bool **bDebug**
- long **lSensorCount**
- long **lDirectionCount**
- long **lFilterSize**
- long **lFFTSize**
- long **lDecimation**
- long **lSpectSize**
- long **lMinBin**
- long **lMaxBin**
- long **lNewData**
- long **lOldData**
- float **fOverlap**
- GDT **fFreqRes**
- GDT **fIntTime**
- GDT * **fpWinFunc**
- GDT * **fpExtBuf**
- std::vector< GDT * > **vfpProcBuf**
- std::vector< GDT * > **vfpPrevBuf**
- std::vector< GDT * > **vfpRNBuf**
- std::vector< GCDT * > **vspSpect**
- clDSPAlloc **WinFuncBuf**
- clDSPAlloc **ExtBuf**
- std::vector< clDSPAlloc > **vProcBuf**
- std::vector< clDSPAlloc > **vPrevBuf**
- std::vector< clDSPAlloc > **vRNBuf**
- std::vector< clDSPAlloc > **vSpectBuf**
- std::vector< clRecDecimator * > **vDecimator**
- **clRemoveNoise BNER**

7.93.1 Detailed Description

Spectrum based direction finding for a line array.

This is 100% my (Jussi Laako) own algorithm. At least I haven't seen this anywhere before and I'm not aware if such exists anywhere else.

Definition at line 68 of file SpectDirLine.hh.

7.93.2 Constructor & Destructor Documentation

7.93.2.1 clSpectDirLine::clSpectDirLine (long, GDT, GDT, double, long, int, long, long, GDT, bool)

Constructor.

Parameters:

lSensors Number of sensors in array

fSensorSpacing Sensor distance (m)

fSoundSpeed Speed of sound (m/s)

dSampleRate Samplerate

lFiltSize Filter size (points)

iFilterType Type of decimation filter

lWindowSize Window size (points)

lDirCount Number of sectors

fIntTimeReq Integration time request (s)

bEnableDebug Enable debug?

Definition at line 106 of file SpectDirLine.cc.

References bDebug, clSpectDir::DSP, fFreqRes, fIntTime, fOverlap, fpWinFunc, GCDT, GDT, clSpectDir::GetArrayFrequency(), lDecimation, lDirectionCount, lFFTSize, lFilterSize, lMaxBin, lNewData, lOldData, lSensorCount, lSpectSize, clSpectDir::SetDirectionCount(), clSpectDir::SetSensorSpacing(), clSpectDir::SetSoundSpeed(), vDecimator, vfpPrevBuf, vfpProcBuf, vfpRNBuf, vPrevBuf, vProcBuf, vRNBuf, vSpectBuf, vspSpect, and WinFuncBuf.

7.93.2.2 clSpectDirLine::~clSpectDirLine ()

Definition at line 194 of file SpectDirLine.cc.

References vDecimator.

7.93.3 Member Function Documentation

7.93.3.1 void clSpectDirLine::Calculate (int, stpSpectDirRN) `[private]`

Definition at line 29 of file SpectDirLine.cc.

References BNER, clRemoveNoise::Diff(), clSpectDir::DSP, _stSpectDirRN::fAlpha, fFreqRes, fpWinFunc, GCDT, GDT, clRemoveNoise::InvDiff(), _stSpectDirRN::iType, lFFTSize, _stSpectDirRN::lGapLength, lMaxBin, _stSpectDirRN::lMeanLength, lMinBin, lSensorCount, lSpectSize, clRemoveNoise::OTA(), SDL_BNER_DIFF, SDL_BNER_IDIFF, SDL_BNER_NONE, SDL_BNER_OTA,

SDL_BNER_TPSW, SDL_SCALE_LIN, SDL_SCALE_LOG, clSpectDir::SetDirection(), clRemoveNoise::TPSW(), vfpProcBuf, vfpRNBuf, and vspSpect.

Referenced by GetResults().

7.93.3.2 void clSpectDirLine::PutData (const GDT *, long, long, long)

Put data into input data FIFO.

Parameters:

fpInputData Input data

lDataCount Number of samples (total)

lStartCh Starting channel index

lChCount Number of channels (total)

Definition at line 204 of file SpectDirLine.cc.

References clSpectDir::DSP, ExtBuf, fpExtBuf, GDT, lSensorCount, and vDecimator.

Referenced by clDirection2::ProcessLoop().

7.93.3.3 bool clSpectDirLine::GetResults (GDT *, GDT, int, stpSpectDirRN)

Get results.

$$V_f(x) = \sqrt{\left(\Re F(x)_L{}^2 + \Im F(x)_L{}^2\right) + \left(\Re F(x)_R{}^2 + \Im F(x)_R{}^2\right)}$$

$$z_c(x) = F(x)_L F(x)_R^*$$

$$\Delta\varphi_f(x) = \arctan\left(\frac{\Im z_c(x)}{\Re z_c(x)}\right)$$

Parameters:

fpResults Results vector

fLowFreqLimit Lower frequency limit (Hz)

iScaling Scaling type

spRemoveNoise Noise removal parameters

Returns:

Results available?

Definition at line 222 of file SpectDirLine.cc.

References Calculate(), clSpectDir::DSP, fFreqRes, GDT, clSpectDir::GetDirections(), lFFTSize, lMaxBin, lMinBin, lNewData, lOldData, lSensorCount, vDecimator, vfpPrevBuf, and vfpProcBuf.

Referenced by clDirection2::ProcessLoop().

7.93.3.4 void clSpectDirLine::ResetResults ()

Reset results.

Definition at line 256 of file SpectDirLine.cc.

References clSpectDir::ResetDirections().

Referenced by clDirection2::ProcessLoop().

7.93.3.5 GDT clSpectDirLine::GetIntegrationTime () `[inline]`

Get integration time (seconds).

Returns:
Integration time (s)

Definition at line 149 of file SpectDirLine.hh.

References fIntTime, and GDT.

Referenced by clDirection2::ProcessLoop().

7.93.4 Member Data Documentation

7.93.4.1 bool clSpectDirLine::bDebug `[private]`

Definition at line 70 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.2 long clSpectDirLine::lSensorCount `[private]`

Definition at line 71 of file SpectDirLine.hh.

Referenced by Calculate(), clSpectDirLine(), GetResults(), and PutData().

7.93.4.3 long clSpectDirLine::lDirectionCount `[private]`

Definition at line 72 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.4 long clSpectDirLine::lFilterSize `[private]`

Definition at line 73 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.5 long clSpectDirLine::lFFTSize [private]

Definition at line 74 of file SpectDirLine.hh.

Referenced by Calculate(), clSpectDirLine(), and GetResults().

7.93.4.6 long clSpectDirLine::lDecimation [private]

Definition at line 75 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.7 long clSpectDirLine::lSpectSize [private]

Definition at line 76 of file SpectDirLine.hh.

Referenced by Calculate(), and clSpectDirLine().

7.93.4.8 long clSpectDirLine::lMinBin [private]

Definition at line 77 of file SpectDirLine.hh.

Referenced by Calculate(), and GetResults().

7.93.4.9 long clSpectDirLine::lMaxBin [private]

Definition at line 78 of file SpectDirLine.hh.

Referenced by Calculate(), clSpectDirLine(), and GetResults().

7.93.4.10 long clSpectDirLine::lNewData [private]

Definition at line 79 of file SpectDirLine.hh.

Referenced by clSpectDirLine(), and GetResults().

7.93.4.11 long clSpectDirLine::lOldData [private]

Definition at line 80 of file SpectDirLine.hh.

Referenced by clSpectDirLine(), and GetResults().

7.93.4.12 float clSpectDirLine::fOverlap [private]

Definition at line 81 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.13 GDT clSpectDirLine::fFreqRes [private]

Definition at line 82 of file SpectDirLine.hh.

Referenced by Calculate(), clSpectDirLine(), and GetResults().

7.93.4.14 GDT clSpectDirLine::fIntTime [private]

Definition at line 83 of file SpectDirLine.hh.

Referenced by clSpectDirLine(), and GetIntegrationTime().

7.93.4.15 GDT* clSpectDirLine::fpWinFunc [private]

Definition at line 84 of file SpectDirLine.hh.

Referenced by Calculate(), and clSpectDirLine().

7.93.4.16 GDT* clSpectDirLine::fpExtBuf [private]

Definition at line 85 of file SpectDirLine.hh.

Referenced by PutData().

7.93.4.17 std::vector<GDT *> clSpectDirLine::vfpProcBuf [private]

Definition at line 86 of file SpectDirLine.hh.

Referenced by Calculate(), clSpectDirLine(), and GetResults().

7.93.4.18 std::vector<GDT *> clSpectDirLine::vfpPrevBuf [private]

Definition at line 87 of file SpectDirLine.hh.

Referenced by clSpectDirLine(), and GetResults().

7.93.4.19 std::vector<GDT *> clSpectDirLine::vfpRNBuf [private]

Definition at line 88 of file SpectDirLine.hh.

Referenced by Calculate(), and clSpectDirLine().

7.93.4.20 std::vector<GCDT *> clSpectDirLine::vspSpect [private]

Definition at line 89 of file SpectDirLine.hh.

Referenced by Calculate(), and clSpectDirLine().

7.93.4.21 clDSPAlloc clSpectDirLine::WinFuncBuf [private]

Definition at line 90 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.22 clDSPAlloc clSpectDirLine::ExtBuf [private]

Definition at line 91 of file SpectDirLine.hh.

Referenced by PutData().

7.93.4.23 std::vector<clDSPAlloc> clSpectDirLine::vProcBuf [private]

Definition at line 92 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.24 std::vector<clDSPAlloc> clSpectDirLine::vPrevBuf [private]

Definition at line 93 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.25 std::vector<clDSPAlloc> clSpectDirLine::vRNBuf [private]

Definition at line 94 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.26 std::vector<clDSPAlloc> clSpectDirLine::vSpectBuf [private]

Definition at line 95 of file SpectDirLine.hh.

Referenced by clSpectDirLine().

7.93.4.27 std::vector<clRecDecimator *> clSpectDirLine::vDecimator [private]

Definition at line 96 of file SpectDirLine.hh.

Referenced by clSpectDirLine(), GetResults(), PutData(), and ~clSpectDirLine().

7.93.4.28 clRemoveNoise clSpectDirLine::BNER [private]

Definition at line 97 of file SpectDirLine.hh.

Referenced by Calculate().

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

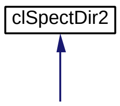
- **SpectDirLine.hh**
- **SpectDirLine.cc**

7.94 clSpectDirLine2 Class Reference

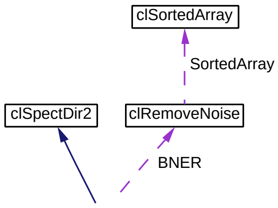
Spectrum based direction finding for line array.

`#include <SpectDirLine2.hh>`

Inheritance diagram for clSpectDirLine2:



Collaboration diagram for clSpectDirLine2:



Public Member Functions

- **clSpectDirLine2** (long, GDT, GDT, double, long, int, long, GDT, bool)

  Constructor.

- **~clSpectDirLine2** ()
- void **PutData** (const GDT *, long, long, long)

  Put data into input FIFO.

- bool **GetResults** (GDT *, GDT *, GDT, int, **stpSpectDir2RN**)

  Get copy of results.

- GDT * **GetLevels** ()

  Get pointer to level results.

- GDT * **GetDirections** ()

  Get pointer to direction results.

- GDT **GetIntegrationTime** ()

  Get integration time (seconds).

- GDT **GetFreqResolution** ()
  Get frequency resolution (Hz).
- long **GetMinBin** ()
  Get minimum used spectrum bin.
- long **GetMaxBin** ()
  Get maximum used spectrum bin.
- long **GetResultCount** ()
  Get result count.

Private Member Functions

- void **Calculate** (int, **stpSpectDir2RN**)

Private Attributes

- bool **bDebug**
- long **lSensorCount**
- long **lFilterSize**
- long **lFFTSize**
- long **lDecimation**
- long **lSpectSize**
- long **lMinBin**
- long **lMaxBin**
- long **lNewData**
- long **lOldData**
- float **fOverlap**
- GDT **fFreqRes**
- GDT **fIntTime**
- GDT * **fpWinFunc**
- GDT * **fpLevRes**
- GDT * **fpDirRes**
- GDT * **fpExtBuf**
- std::vector< GDT * > **vfpProcBuf**
- std::vector< GDT * > **vfpPrevBuf**
- std::vector< GDT * > **vfpRNBuf**
- std::vector< GCDT * > **vspSpect**
- clDSPAlloc **WinFuncBuf**
- clDSPAlloc **LevResBuf**
- clDSPAlloc **DirResBuf**
- clDSPAlloc **ExtBuf**
- std::vector< clDSPAlloc > **vProcBuf**

- std::vector< clDSPAlloc > **vPrevBuf**
- std::vector< clDSPAlloc > **vRNBuf**
- std::vector< clDSPAlloc > **vSpectBuf**
- std::vector< clRecDecimator * > **vDecimator**
- **clRemoveNoise BNER**

7.94.1 Detailed Description

Spectrum based direction finding for line array.

This is used by **clDirection3**(p. 185).

Note:

See **clSpectDirLine**(p. 542) for details!

Definition at line 63 of file SpectDirLine2.hh.

7.94.2 Constructor & Destructor Documentation

7.94.2.1 clSpectDirLine2::clSpectDirLine2 (long, GDT, GDT, double, long, int, long, GDT, bool)

Constructor.

Parameters:

lSensors Number of sensors in array

fSensorSpacing Sensor spacing (m)

fSoundSpeed Speed of sound (m/s)

dSampleRate Samplerate

lFiltSize Size of filter (points)

iFilterType Type of decimation filter

lWindowSize Size of window (points)

fIntTimeReq Integration time request (s)

bEnableDebug Enable debug?

Definition at line 114 of file SpectDirLine2.cc.

References bDebug, DirResBuf, clSpectDir2::DSP, fFreqRes, fIntTime, f-Overlap, fpDirRes, fpLevRes, fpWinFunc, GCDT, GDT, clSpectDir2::Get-ArrayFrequency(), lDecimation, LevResBuf, lFFTSize, lFilterSize, lMax-Bin, lNewData, lOldData, lSensorCount, lSpectSize, clSpectDir2::SetSensor-Spacing(), clSpectDir2::SetSoundSpeed(), vDecimator, vfpPrevBuf, vfpProc-Buf, vfpRNBuf, vPrevBuf, vProcBuf, vRNBuf, vSpectBuf, vspSpect, and Win-FuncBuf.

7.94.2.2 clSpectDirLine2::~clSpectDirLine2 ()

Definition at line 201 of file SpectDirLine2.cc.

References vDecimator.

7.94.3 Member Function Documentation

7.94.3.1 void clSpectDirLine2::Calculate (int, stpSpectDir2RN) [private]

Definition at line 30 of file SpectDirLine2.cc.

References BNER, clRemoveNoise::Diff(), clSpectDir2::DSP, _stSpectDir2RN::f-Alpha, fFreqRes, fpDirRes, fpLevRes, fpWinFunc, GCDT, GDT, clSpect-Dir2::GetDirection(), clRemoveNoise::InvDiff(), _stSpectDir2RN::iType, lFFTSize, _stSpectDir2RN::lGapLength, lMaxBin, _stSpectDir2RN::lMean-Length, lMinBin, lSensorCount, lSpectSize, clRemoveNoise::OTA(), SDL2_-BNER_DIFF, SDL2_BNER_IDIFF, SDL2_BNER_NONE, SDL2_BNER_OTA, SDL2_BNER_TPSW, SDL2_SCALE_LIN, SDL2_SCALE_LOG, clRemove-Noise::TPSW(), vfpProcBuf, vfpRNBuf, and vspSpect.

Referenced by GetResults().

7.94.3.2 void clSpectDirLine2::PutData (const GDT *, long, long, long)

Put data into input FIFO.

Parameters:

fpInputData Input data vector

lSampleCount Number of sample in input vector (total)

lStartCh Starting channel index

lChCount Channel count, total

Definition at line 211 of file SpectDirLine2.cc.

References clSpectDir2::DSP, ExtBuf, fpExtBuf, GDT, lSensorCount, and v-Decimator.

Referenced by clDirection3::ProcessLoop().

7.94.3.3 bool clSpectDirLine2::GetResults (GDT *, GDT *, GDT, int, stpSpectDir2RN)

Get copy of results.

Parameters:

fpLevelResults Levels vector (can be NULL)

fpDirResults Directions vector (can be NULL)

fLowFreqLimit Low frequency limit (Hz)

iScaling Scaling type

spRemoveNoise Noise removal parameters

Returns:

Results available?

Definition at line 228 of file SpectDirLine2.cc.

References Calculate(), clSpectDir2::DSP, fFreqRes, fpDirRes, fpLevRes, GDT, lFFTSize, lMaxBin, lMinBin, lNewData, lOldData, lSensorCount, lSpectSize, vDecimator, vfpPrevBuf, and vfpProcBuf.

Referenced by clDirection3::ProcessLoop().

7.94.3.4 GDT* clSpectDirLine2::GetLevels () `[inline]`

Get pointer to level results.

Returns:

Pointer to level results

Definition at line 138 of file SpectDirLine2.hh.

References fpLevRes, and GDT.

Referenced by clDirection3::ProcessLoop().

7.94.3.5 GDT* clSpectDirLine2::GetDirections () `[inline]`

Get pointer to direction results.

Returns:

Pointer to direction results

Definition at line 144 of file SpectDirLine2.hh.

References fpDirRes, and GDT.

Referenced by clDirection3::ProcessLoop().

7.94.3.6 GDT clSpectDirLine2::GetIntegrationTime () `[inline]`

Get integration time (seconds).

Returns:

Integration time (s)

Definition at line 150 of file SpectDirLine2.hh.

References fIntTime, and GDT.

Referenced by clDirection3::ProcessLoop().

7.94.3.7 GDT clSpectDirLine2::GetFreqResolution () [inline]

Get frequency resolution (Hz).

Returns:
Frequency resolution (Hz)

Definition at line 156 of file SpectDirLine2.hh.

References fFreqRes, and GDT.

Referenced by clDirection3::ProcessLoop().

7.94.3.8 long clSpectDirLine2::GetMinBin () [inline]

Get minimum used spectrum bin.

Returns:
Index to lowest used bin

Definition at line 162 of file SpectDirLine2.hh.

References lMinBin.

Referenced by clDirection3::ProcessLoop().

7.94.3.9 long clSpectDirLine2::GetMaxBin () [inline]

Get maximum used spectrum bin.

Returns:
Index to highest used bin

Definition at line 168 of file SpectDirLine2.hh.

References lMaxBin.

Referenced by clDirection3::ProcessLoop().

7.94.3.10 long clSpectDirLine2::GetResultCount () [inline]

Get result count.

Returns:
Result vector length

Definition at line 174 of file SpectDirLine2.hh.

References lSpectSize.

Referenced by clDirection3::ProcessLoop().

7.94.4 Member Data Documentation

7.94.4.1 bool clSpectDirLine2::bDebug `[private]`

Definition at line 65 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.2 long clSpectDirLine2::lSensorCount `[private]`

Definition at line 66 of file SpectDirLine2.hh.

Referenced by Calculate(), clSpectDirLine2(), GetResults(), and PutData().

7.94.4.3 long clSpectDirLine2::lFilterSize `[private]`

Definition at line 67 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.4 long clSpectDirLine2::lFFTSize `[private]`

Definition at line 68 of file SpectDirLine2.hh.

Referenced by Calculate(), clSpectDirLine2(), and GetResults().

7.94.4.5 long clSpectDirLine2::lDecimation `[private]`

Definition at line 69 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.6 long clSpectDirLine2::lSpectSize `[private]`

Definition at line 70 of file SpectDirLine2.hh.

Referenced by Calculate(), clSpectDirLine2(), GetResultCount(), and GetResults().

7.94.4.7 long clSpectDirLine2::lMinBin `[private]`

Definition at line 71 of file SpectDirLine2.hh.

Referenced by Calculate(), GetMinBin(), and GetResults().

7.94.4.8 long clSpectDirLine2::lMaxBin [private]

Definition at line 72 of file SpectDirLine2.hh.

Referenced by Calculate(), clSpectDirLine2(), GetMaxBin(), and GetResults().

7.94.4.9 long clSpectDirLine2::lNewData [private]

Definition at line 73 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2(), and GetResults().

7.94.4.10 long clSpectDirLine2::lOldData [private]

Definition at line 74 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2(), and GetResults().

7.94.4.11 float clSpectDirLine2::fOverlap [private]

Definition at line 75 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.12 GDT clSpectDirLine2::fFreqRes [private]

Definition at line 76 of file SpectDirLine2.hh.

Referenced by Calculate(), clSpectDirLine2(), GetFreqResolution(), and GetResults().

7.94.4.13 GDT clSpectDirLine2::fIntTime [private]

Definition at line 77 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2(), and GetIntegrationTime().

7.94.4.14 GDT* clSpectDirLine2::fpWinFunc [private]

Definition at line 78 of file SpectDirLine2.hh.

Referenced by Calculate(), and clSpectDirLine2().

7.94.4.15 GDT* clSpectDirLine2::fpLevRes [private]

Definition at line 79 of file SpectDirLine2.hh.

Referenced by Calculate(), clSpectDirLine2(), GetLevels(), and GetResults().

7.94.4.16 GDT* clSpectDirLine2::fpDirRes [private]

Definition at line 80 of file SpectDirLine2.hh.

Referenced by Calculate(), clSpectDirLine2(), GetDirections(), and GetResults().

7.94.4.17 GDT* clSpectDirLine2::fpExtBuf [private]

Definition at line 81 of file SpectDirLine2.hh.

Referenced by PutData().

7.94.4.18 std::vector<GDT *> clSpectDirLine2::vfpProcBuf [private]

Definition at line 82 of file SpectDirLine2.hh.

Referenced by Calculate(), clSpectDirLine2(), and GetResults().

7.94.4.19 std::vector<GDT *> clSpectDirLine2::vfpPrevBuf [private]

Definition at line 83 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2(), and GetResults().

7.94.4.20 std::vector<GDT *> clSpectDirLine2::vfpRNBuf [private]

Definition at line 84 of file SpectDirLine2.hh.

Referenced by Calculate(), and clSpectDirLine2().

7.94.4.21 std::vector<GCDT *> clSpectDirLine2::vspSpect [private]

Definition at line 85 of file SpectDirLine2.hh.

Referenced by Calculate(), and clSpectDirLine2().

7.94.4.22 clDSPAlloc clSpectDirLine2::WinFuncBuf [private]

Definition at line 86 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.23 clDSPAlloc clSpectDirLine2::LevResBuf [private]

Definition at line 87 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.24 clDSPAlloc clSpectDirLine2::DirResBuf [private]

Definition at line 88 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.25 clDSPAlloc clSpectDirLine2::ExtBuf [private]

Definition at line 89 of file SpectDirLine2.hh.

Referenced by PutData().

7.94.4.26 std::vector<clDSPAlloc> clSpectDirLine2::vProcBuf [private]

Definition at line 90 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.27 std::vector<clDSPAlloc> clSpectDirLine2::vPrevBuf [private]

Definition at line 91 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.28 std::vector<clDSPAlloc> clSpectDirLine2::vRNBuf [private]

Definition at line 92 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.29 std::vector<clDSPAlloc> clSpectDirLine2::vSpectBuf [private]

Definition at line 93 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2().

7.94.4.30 std::vector<clRecDecimator *> clSpectDirLine2::vDecimator [private]

Definition at line 94 of file SpectDirLine2.hh.

Referenced by clSpectDirLine2(), GetResults(), PutData(), and ~clSpectDirLine2().

7.94.4.31 clRemoveNoise clSpectDirLine2::BNER [private]

Definition at line 95 of file SpectDirLine2.hh.

Referenced by Calculate().

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

- **SpectDirLine2.hh**
- **SpectDirLine2.cc**

7.95 _stSpectDirRN Struct Reference

SpectDir: Noise removal parameters.

`#include <SpectDir.hh>`

Public Attributes

- int **iType**
- GDT **fAlpha**
- long **lMeanLength**
- long **lGapLength**

7.95.1 Detailed Description

SpectDir: Noise removal parameters.

Definition at line 38 of file SpectDir.hh.

7.95.2 Member Data Documentation

7.95.2.1 int _stSpectDirRN::iType

Definition at line 40 of file SpectDir.hh.

Referenced by clSpectDirLine::Calculate(), clSpectDirDipole::Calculate(), and clDirection2::ProcessLoop().

7.95.2.2 GDT _stSpectDirRN::fAlpha

Definition at line 41 of file SpectDir.hh.

Referenced by clSpectDirLine::Calculate(), clSpectDirDipole::Calculate(), and clDirection2::ProcessLoop().

7.95.2.3 long _stSpectDirRN::lMeanLength

Definition at line 42 of file SpectDir.hh.

Referenced by clSpectDirLine::Calculate(), clSpectDirDipole::Calculate(), and clDirection2::ProcessLoop().

7.95.2.4 long _stSpectDirRN::lGapLength

Definition at line 43 of file SpectDir.hh.

Referenced by clSpectDirLine::Calculate(), clSpectDirDipole::Calculate(), and clDirection2::ProcessLoop().

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

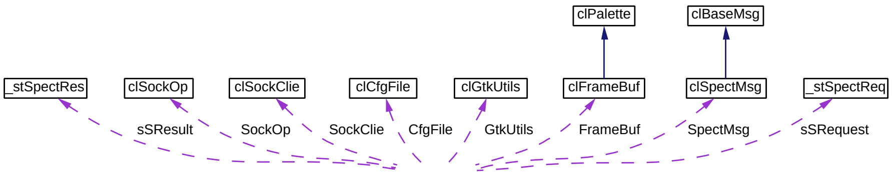
- **SpectDir.hh**

7.96 clSpectGUI Class Reference

Transient spectrum GUI.

`#include <GUISpect.hh>`

Collaboration diagram for clSpectGUI:



Public Member Functions

- **clSpectGUI** (int ∗, char ∗∗∗)
- ∼**clSpectGUI** ()
- int **Exec** ()
- gint **OnDelete** (GtkWidget ∗, GdkEventAny ∗)
- void **OnHideToggled** (GtkToggleButton ∗, gpointer)
- gint **OnConnectClick** (GtkWidget ∗, gpointer)
- void **OnFreezeToggled** (GtkToggleButton ∗, gpointer)
- gint **OnPaletteActivate** (GtkWidget ∗, gpointer)
- gint **OnMotionSgram** (GtkWidget ∗, GdkEventMotion ∗)
- gint **OnMotionSpect** (GtkWidget ∗, GdkEventMotion ∗)
- gint **OnExposeSgram** (GtkWidget ∗, GdkEventExpose ∗)
- gint **OnExposeSpect** (GtkWidget ∗, GdkEventExpose ∗)
- gboolean **OnConfigure** (GtkWidget ∗, GdkEventConfigure ∗, gpointer)
- void **OnSizeAllocate** (GtkWidget ∗, GtkAllocation ∗, gpointer)
- void **GdkInput** (gpointer, gint, GdkInputCondition)
- void **OnSaveClicks** (GtkButton ∗, gpointer)

Private Member Functions

- bool **Build** ()
- bool **BuildTable1** ()
- bool **BuildTable2** ()
- bool **BuildTable3** ()
- bool **BuildTableGram** ()
- bool **BuildTableSpect** ()
- bool **BuildDrawingPrims** ()
- void **FreeDrawingPrims** ()
- bool **ConnectSignals** ()
- int **GetGramHeight** ()

- bool **ParseServerStr** (char ∗, int ∗, const char ∗)
- bool **InitConnection** (const char ∗, int)
- void **GetSettings** ()
- bool **SendSettings** ()
- void **ReConfigDisplay** ()
- void **DrawSpectrogram** ()
- void **DrawSpectrum** ()
- void **PrintRealTime** ()
- long **CountBands** (long)
- void **SaveInfo** (const char ∗)

Private Attributes

- bool **bRun**
- bool **bConnected**
- bool **bFreezed**
- bool **bReConfig**
- int **iSockH**
- int **iRcvMsgSize**
- int **iFit**
- int **iPalette**
- int **iGramW**
- int **iGramH**
- int **iSpectW**
- int **iSpectH**
- int **iScaleFactor**
- int **iSpectPoints**
- int **iClips**
- int **iTIFFCompression**
- int **iJPEGQuality**
- int **iBeamCount**
- float **fGramX**
- float **fGramY**
- float **fSpectX**
- float **fSpectY**
- char **cpGramXTime** [20]
- char ∗ **cpRcvMsgBuf**
- GDT ∗ **fpSpect**
- GDT ∗ **fpIntSpect**
- gint **giGdkTag**
- GList ∗ **glServers**
- GtkWidget ∗ **gwWindow**
- GtkWidget ∗ **gwVBox**
- GtkWidget ∗ **gwCBHide**
- GtkWidget ∗ **gwVPaned**
- GtkWidget ∗ **gwScrolledW1**

- GtkWidget * **gwScrolledW2**
- GtkWidget * **gwStatusBar**
- guint **guCtxtSB**
- GtkWidget * **gwTable1**
- GtkWidget * **gwLServer**
- GtkWidget * **gwCServer**
- GtkWidget * **gwLChannel**
- GtkObject * **goAChannel**
- GtkWidget * **gwSBChannel**
- GtkWidget * **gwBConnect**
- GtkWidget * **gwBDisconnect**
- GtkWidget * **gwCBFreeze**
- GtkWidget * **gwTable2**
- GtkWidget * **gwLType**
- GtkWidget * **gwOMType**
- GtkWidget * **gwMType**
- GtkWidget * **gwaMIType** [SGUI_TYPE_ITEMS]
- GtkWidget * **gwLWindow**
- GtkWidget * **gwOMWindow**
- GtkWidget * **gwMWindow**
- GtkWidget * **gwaMIWindow** [SGUI_WINDOW_ITEMS]
- GtkWidget * **gwLWindowParam**
- GtkWidget * **gwEWindowParam**
- GtkWidget * **gwLWindowLen**
- GtkWidget * **gwOMWindowLen**
- GtkWidget * **gwMWindowLen**
- GtkWidget * **gwaMIWindowLen** [SGUI_WINLEN_ITEMS]
- GtkWidget * **gwLLowFreq**
- GtkWidget * **gwELowFreq**
- GtkWidget * **gwLHighFreq**
- GtkWidget * **gwEHighFreq**
- GtkWidget * **gwLGain**
- GtkWidget * **gwEGain**
- GtkWidget * **gwLSlope**
- GtkWidget * **gwESlope**
- GtkWidget * **gwLOverlap**
- GtkWidget * **gwEOverlap**
- GtkWidget * **gwCBLinear**
- GtkWidget * **gwCBNormalize**
- GtkWidget * **gwTable3**
- GtkWidget * **gwLRemoveNoise**
- GtkWidget * **gwOMRemoveNoise**
- GtkWidget * **gwMRemoveNoise**
- GtkWidget * **gwaMIRemoveNoise** [SGUI_REMOVE_NOISE_ITEMS]
- GtkWidget * **gwLAlpha**
- GtkWidget * **gwEAlpha**

- GtkWidget * **gwLMeanLength**
- GtkWidget * **gwEMeanLength**
- GtkWidget * **gwLGapLength**
- GtkWidget * **gwEGapLength**
- GtkWidget * **gwLDynRange**
- GtkWidget * **gwEDynRange**
- GtkWidget * **gwLPalette**
- GtkWidget * **gwOMPalette**
- GtkWidget * **gwMPalette**
- GtkWidget * **gwaMIPalette** [SGUI_PALETTE_ITEMS]
- GtkWidget * **gwBSave**
- GtkWidget * **gwFSSave**
- GtkWidget * **gwTableGram**
- GtkWidget * **gwHRTime**
- GtkWidget * **gwVRFreq**
- GtkWidget * **gwDASpectrogram**
- GtkWidget * **gwTableSpect**
- GtkWidget * **gwHRFreq**
- GtkWidget * **gwVRLevel**
- GtkWidget * **gwDASpectrum**
- GdkGC * **ggcGramBG**
- GdkGC * **ggcGramFG**
- GdkGC * **ggcSpectBG**
- GdkGC * **ggcSpectFG**
- GdkCursor * **gcCrossHair**
- **stSpectReq sSRequest**
- **stSpectRes sSResult**
- clAlloc **RcvMsgBuf**
- clAlloc **SpectBuf**
- clAlloc **IntSpectBuf**
- **clCfgFile * CfgFile**
- clDSPOp **DSP**
- **clGtkUtils GtkUtils**
- **clSockClie SockClie**
- **clSockOp * SockOp**
- **clSpectMsg SpectMsg**
- **clFrameBuf FrameBuf**

7.96.1 Detailed Description

Transient spectrum GUI.

Definition at line 94 of file GUISpect.hh.

7.96.2 Constructor & Destructor Documentation

7.96.2.1 clSpectGUI::clSpectGUI (int *, char ***)

Definition at line 178 of file GUISpect.cc.

References bConnected, bFreezed, bRun, CfgFile, cpRcvMsgBuf, cpWindowTxt, fpSpect, clCfgFile::GetInt(), glServers, iBeamCount, iFit, iJPEGQuality, iPalette, iTIFFCompression, SGUI_CFGFILE, SGUI_VER_MAJ, SGUI_VER_MIN, and SGUI_VER_PL.

7.96.2.2 clSpectGUI::~clSpectGUI ()

Definition at line 210 of file GUISpect.cc.

References bConnected, CfgFile, and SockOp.

7.96.3 Member Function Documentation

7.96.3.1 bool clSpectGUI::Build () `[private]`

Definition at line 655 of file GUISpect.cc.

References BuildDrawingPrims(), BuildTable1(), BuildTable2(), BuildTable3(), BuildTableGram(), BuildTableSpect(), CfgFile, cpWindowTxt, fGramX, fGramY, _stSpectRes::fPeakLevel, FrameBuf, fSpectX, fSpectY, gcCrossHair, clFrameBuf::GetFBPtr(), clCfgFile::GetInt(), clFrameBuf::GetNumColors(), clFrameBuf::GetPalPtr(), guCtxtSB, gwCBHide, gwDASpectrogram, gwDASpectrum, gwScrolledW1, gwScrolledW2, gwStatusBar, gwVBox, gwVPaned, gwWindow, iFit, iGramH, iGramW, _stSpectRes::lLength, clFrameBuf::SetSize(), SGUI_CONV_MAXLEN, SGUI_DEF_HEIGHT, SGUI_DEF_WIDTH, SGUI_PADDING, and sSResult.

Referenced by Exec().

7.96.3.2 bool clSpectGUI::BuildTable1 () `[private]`

Definition at line 803 of file GUISpect.cc.

References clGtkUtils::ComboListFromFile(), cpaLChannelTxt, cpBConnectTxt, cpBDisconnectTxt, cpCBFreezeTxt, cpLServerTxt, glServers, goAChannel, GtkUtils, gwBConnect, gwBDisconnect, gwCBFreeze, gwCServer, gwLChannel, gwLServer, gwSBChannel, gwTable1, gwVBox, iBeamCount, SGUI_CH_LOWER, SGUI_CH_UPPER, SGUI_HOSTFILE, SGUI_PADDING, and SGUI_SERVER_MAXLEN.

Referenced by Build().

7.96.3.3 bool clSpectGUI::BuildTable2 () [private]

Definition at line 898 of file GUISpect.cc.

References clGtkUtils::BuildOptionMenu(), cpaLTypeMenu, cpaLWindowLenMenu, cpaLWindowMenu, cpLGainTxt, cpLHighFreqTxt, cpLLinearTxt, cpLLowFreqTxt, cpLNormalizeTxt, cpLOverlapTxt, cpLSlopeTxt, cpLTypeTxt, cpLWindowLenTxt, cpLWindowParamTxt, cpLWindowTxt, GtkUtils, gwaMIType, gwaMIWindow, gwaMIWindowLen, gwCBLinear, gwCBNormalize, gwEGain, gwEHighFreq, gwELowFreq, gwEOverlap, gwESlope, gwEWindowParam, gwLGain, gwLHighFreq, gwLLowFreq, gwLOverlap, gwLSlope, gwLType, gwLWindow, gwLWindowLen, gwLWindowParam, gwMType, gwMWindow, gwMWindowLen, gwOMType, gwOMWindow, gwOMWindowLen, gwTable2, gwVBox, SGUI_PADDING, SGUI_TYPE_ITEMS, SGUI_WINDOW_ITEMS, and SGUI_WINLEN_ITEMS.

Referenced by Build().

7.96.3.4 bool clSpectGUI::BuildTable3 () [private]

Definition at line 1079 of file GUISpect.cc.

References clGtkUtils::BuildOptionMenu(), cpaLPaletteMenu, cpaLRemoveNoiseMenu, cpBSaveTxt, cpFSSaveTxt, cpLAlphaTxt, cpLDynRangeTxt, cpLGapLengthTxt, cpLMeanLengthTxt, cpLPaletteTxt, cpLRemoveNoiseTxt, GtkUtils, gwaMIPalette, gwaMIRemoveNoise, gwBSave, gwEAlpha, gwEDynRange, gwEGapLength, gwEMeanLength, gwFSSave, gwLAlpha, gwLDynRange, gwLGapLength, gwLMeanLength, gwLPalette, gwLRemoveNoise, gwMPalette, gwMRemoveNoise, gwOMPalette, gwOMRemoveNoise, gwTable3, gwVBox, iPalette, SGUI_PADDING, SGUI_PALETTE_ITEMS, and SGUI_REMOVE_NOISE_ITEMS.

Referenced by Build().

7.96.3.5 bool clSpectGUI::BuildTableGram () [private]

Definition at line 1209 of file GUISpect.cc.

References gwDASpectrogram, gwHRTime, gwScrolledW1, gwTableGram, and gwVRFreq.

Referenced by Build().

7.96.3.6 bool clSpectGUI::BuildTableSpect () [private]

Definition at line 1253 of file GUISpect.cc.

References gwDASpectrum, gwHRFreq, gwScrolledW2, gwTableSpect, and gwVRLevel.

Referenced by Build().

7.96.3.7 bool clSpectGUI::BuildDrawingPrims () [private]

Definition at line 1297 of file GUISpect.cc.

References CfgFile, FrameBuf, gcCrossHair, clCfgFile::GetInt(), ggcGramBG, ggcGramFG, ggcSpectBG, ggcSpectFG, gwDASpectrogram, gwDASpectrum, clFrameBuf::PalGenBW(), clFrameBuf::PalGenDir(), clFrameBuf::PalGen-Green(), clFrameBuf::PalGenHSV(), clFrameBuf::PalGenLight(), clFrame-Buf::PalGenTemp(), SGUI_GRAM_BG, SGUI_GRAM_FG, SGUI_PAL_BW, SGUI_PAL_DIR, SGUI_PAL_GREEN, SGUI_PAL_HSV, SGUI_PAL_LIGHT, SGUI_PAL_TEMP, SGUI_SPECT_BG, and SGUI_SPECT_FG.

Referenced by Build().

7.96.3.8 void clSpectGUI::FreeDrawingPrims () [private]

Definition at line 1359 of file GUISpect.cc.

References gcCrossHair, ggcGramBG, ggcGramFG, ggcSpectBG, and ggcSpect-FG.

Referenced by Exec().

7.96.3.9 bool clSpectGUI::ConnectSignals () [private]

Definition at line 1369 of file GUISpect.cc.

References clGtkUtils::ConnectMotionEvent(), GtkUtils, gwaMIPalette, gwBConnect, gwBDisconnect, gwBSave, gwCBFreeze, gwCBHide, gw-DASpectrogram, gwDASpectrum, gwFSSave, gwHRFreq, gwHRTime, gwScrolledW2, gwVRFreq, gwVRLevel, gwWindow, SGUI_PALETTE_-ITEMS, WrapOnConnectClick(), WrapOnDelete(), WrapOnExposeSgram(), WrapOnExposeSpect(), WrapOnFreezeToggled(), WrapOnHideToggled(), WrapOnMotionSgram(), WrapOnMotionSpect(), WrapOnPaletteActivate(), WrapOnSaveClicks(), and WrapOnSizeAllocate().

Referenced by Exec().

7.96.3.10 int clSpectGUI::GetGramHeight () [private]

Definition at line 1419 of file GUISpect.cc.

References gwHRTime, and gwScrolledW1.

7.96.3.11 bool clSpectGUI::ParseServerStr (char *, int *, const char *) [private]

Definition at line 1425 of file GUISpect.cc.

References SGUI_SERVER_MAXLEN.

Referenced by OnConnectClick().

7.96.3.12 bool clSpectGUI::InitConnection (const char *, int) [private]

Definition at line 1448 of file GUISpect.cc.

References bConnected, clSockClie::Connect(), GetSettings(), giGdkTag, GLOBAL_HEADER_LEN, iClips, iSockH, SendSettings(), SGUI_REQ_PROC, SockClie, SockOp, WrapGdkInput(), and clSockOp::WriteN().

Referenced by OnConnectClick().

7.96.3.13 void clSpectGUI::GetSettings () [private]

Definition at line 1469 of file GUISpect.cc.

References _stSpectReq::bLinear, _stSpectReq::bNormalize, cpaLWindowLenMenu, cpRcvMsgBuf, _stSpectReq::fAlpha, _stSpectReq::fDynRange, _stSpectReq::fGain, fpSpect, _stSpectReq::fSlope, _stSpectReq::fWinParam, GDT, GLOBAL_HEADER_LEN, GtkUtils, gwaMIRemoveNoise, gwaMIType, gwaMIWindow, gwaMIWindowLen, gwCBLinear, gwCBNormalize, gwEAlpha, gwEDynRange, gwEGain, gwEGapLength, gwEHighFreq, gwELowFreq, gwEMeanLength, gwEOverlap, gwESlope, gwEWindowParam, gwOMRemoveNoise, gwOMType, gwOMWindow, gwOMWindowLen, gwSBChannel, iBeamCount, _stSpectReq::iChannel, _stSpectReq::iHighFreq, _stSpectReq::iLowFreq, _stSpectReq::iOverlap, iRcvMsgSize, _stSpectReq::iRemoveNoise, _stSpectReq::iType, _stSpectReq::iWindow, _stSpectReq::lGapLength, _stSpectReq::lLength, _stSpectReq::lMeanLength, MSG_SPECT_TYPE_AUTOCORR, MSG_SPECT_TYPE_CEPSTRUM, MSG_SPECT_TYPE_FFT, MSG_SPECT_TYPE_GABOR, MSG_SPECT_TYPE_HANKEL, MSG_SPECT_TYPE_WVD, clGtkUtils::OptionMenuGetActive(), RcvMsgBuf, SGUI_REMOVE_NOISE_ITEMS, SGUI_TYPE_ITEMS, SGUI_WINDOW_ITEMS, SGUI_WINLEN_ITEMS, SpectBuf, and sSRequest.

Referenced by InitConnection().

7.96.3.14 bool clSpectGUI::SendSettings () [private]

Definition at line 1560 of file GUISpect.cc.

References bReConfig, GLOBAL_HEADER_LEN, clSpectMsg::SetRequest(), SockOp, SpectMsg, sSRequest, and clSockOp::WriteN().

Referenced by InitConnection().

7.96.3.15 void clSpectGUI::ReConfigDisplay () [private]

Definition at line 1572 of file GUISpect.cc.

References _stSpectRes::bLinear, bReConfig, _stSpectReq::fDynRange, _stSpectRes::fLineTime, FrameBuf, gwDASpectrogram, gwDASpectrum, gwHRFreq, gwHRTime, gwVRFreq, gwVRLevel, iFit, iGramH, iGramW, _stSpect-

Res::iHighFreq, _stSpectRes::iLowFreq, iSpectH, iSpectPoints, _stSpectRes::l-Length, clFrameBuf::SetSize(), sSRequest, and sSResult.

Referenced by GdkInput(), OnConfigure(), and OnSizeAllocate().

7.96.3.16 void clSpectGUI::DrawSpectrogram () [private]

Definition at line 1642 of file GUISpect.cc.

References clFrameBuf::DrawColumn(), DSP, fpIntSpect, fpSpect, FrameBuf, GDT, clFrameBuf::GetCurPtr(), clFrameBuf::GetRowStride(), ggcGramFG, gwDASpectrogram, gwHRTime, gwScrolledW1, iFit, iGramH, iGramW, Int-SpectBuf, iSpectPoints, SGUI_FIT_AVERAGE, and SGUI_FIT_NEIGHBOR.

Referenced by GdkInput().

7.96.3.17 void clSpectGUI::DrawSpectrum () [private]

Definition at line 1710 of file GUISpect.cc.

References fpSpect, GDT, ggcSpectFG, gwDASpectrum, gwScrolledW2, iSpect-H, and iSpectPoints.

Referenced by GdkInput().

7.96.3.18 void clSpectGUI::PrintRealTime () [private]

Definition at line 1739 of file GUISpect.cc.

References cpGramXTime, fGramX, sSResult, and _stSpectRes::sTimeStamp.

Referenced by GdkInput(), and OnMotionSgram().

7.96.3.19 long clSpectGUI::CountBands (long) [private]

Definition at line 1750 of file GUISpect.cc.

References SPECT_BAND_LIMIT.

7.96.3.20 void clSpectGUI::SaveInfo (const char *) [private]

Definition at line 1767 of file GUISpect.cc.

References FB_TIFF_MODE, _stSpectRes::fLineTime, _stSpectRes::iHighFreq, _stSpectRes::iLowFreq, and sSResult.

Referenced by OnSaveClicks().

7.96.3.21 int clSpectGUI::Exec ()

Definition at line 217 of file GUISpect.cc.

References Build(), ConnectSignals(), and FreeDrawingPrims().

Referenced by main().

7.96.3.22 gint clSpectGUI::OnDelete (GtkWidget *, GdkEventAny *)

Definition at line 235 of file GUISpect.cc.

References bRun.

Referenced by WrapOnDelete().

7.96.3.23 void clSpectGUI::OnHideToggled (GtkToggleButton *, gpointer)

Definition at line 243 of file GUISpect.cc.

References gwTable1, gwTable2, and gwTable3.

Referenced by WrapOnHideToggled().

7.96.3.24 gint clSpectGUI::OnConnectClick (GtkWidget *, gpointer)

Definition at line 260 of file GUISpect.cc.

References bConnected, giGdkTag, gwBConnect, gwCServer, InitConnection(), ParseServerStr(), SGUI_SERVER_MAXLEN, and SockOp.

Referenced by WrapOnConnectClick().

7.96.3.25 void clSpectGUI::OnFreezeToggled (GtkToggleButton *, gpointer)

Definition at line 296 of file GUISpect.cc.

References bFreezed.

Referenced by WrapOnFreezeToggled().

7.96.3.26 gint clSpectGUI::OnPaletteActivate (GtkWidget *, gpointer)

Definition at line 302 of file GUISpect.cc.

References CfgFile, FrameBuf, clCfgFile::GetInt(), gwaMIPalette, clFrameBuf::PalGenBW(), clFrameBuf::PalGenDir(), clFrameBuf::PalGenGreen(), clFrameBuf::PalGenGreen2(), clFrameBuf::PalGenHSV(), clFrameBuf::PalGenLight(), clFrameBuf::PalGenPureGreen(), clFrameBuf::PalGenTemp(), clFrameBuf::PalGenWB(), SGUI_PAL_BW, SGUI_PAL_DIR, SGUI_PAL_GREEN, SGUI_PAL_GREEN2, SGUI_PAL_HSV, SGUI_PAL_LIGHT, SGUI_PAL_PUREGREEN, SGUI_PAL_TEMP, and SGUI_PAL_WB.

Referenced by WrapOnPaletteActivate().

7.96.3.27 gint clSpectGUI::OnMotionSgram (GtkWidget ∗, GdkEventMotion ∗)

Definition at line 374 of file GUISpect.cc.

References cpGramXTime, fGramX, fGramY, _stSpectRes::fLineTime, _stSpectRes::fPeakLevel, fSpectX, fSpectY, guCtxtSB, gwStatusBar, iClips, iGramH, iGramW, _stSpectRes::iHighFreq, _stSpectRes::iLowFreq, PrintRealTime(), SGUI_CONV_MAXLEN, and sSResult.

Referenced by WrapOnMotionSgram().

7.96.3.28 gint clSpectGUI::OnMotionSpect (GtkWidget ∗, GdkEventMotion ∗)

Definition at line 396 of file GUISpect.cc.

References cpGramXTime, _stSpectReq::fDynRange, fGramX, fGramY, _stSpectRes::fPeakLevel, fSpectX, fSpectY, guCtxtSB, gwStatusBar, iClips, _stSpectRes::iHighFreq, _stSpectRes::iLowFreq, iSpectH, iSpectPoints, SGUI_CONV_MAXLEN, sSRequest, and sSResult.

Referenced by WrapOnMotionSpect().

7.96.3.29 gint clSpectGUI::OnExposeSgram (GtkWidget ∗, GdkEventExpose ∗)

Definition at line 416 of file GUISpect.cc.

References bRun, FrameBuf, clFrameBuf::GetCurPtr(), clFrameBuf::GetRowStride(), ggcGramFG, iGramH, and iGramW.

Referenced by WrapOnExposeSgram().

7.96.3.30 gint clSpectGUI::OnExposeSpect (GtkWidget ∗, GdkEventExpose ∗)

Definition at line 436 of file GUISpect.cc.

References bRun, fpSpect, GDT, ggcSpectFG, and iSpectPoints.

Referenced by WrapOnExposeSpect().

7.96.3.31 gboolean clSpectGUI::OnConfigure (GtkWidget *, GdkEventConfigure *, gpointer)

Definition at line 466 of file GUISpect.cc.

References gwDASpectrogram, gwDASpectrum, iFit, iGramH, iGramW, i-SpectH, iSpectW, and ReConfigDisplay().

Referenced by WrapOnConfigure().

7.96.3.32 void clSpectGUI::OnSizeAllocate (GtkWidget *, GtkAllocation *, gpointer)

Definition at line 497 of file GUISpect.cc.

References gwDASpectrogram, gwDASpectrum, gwHRFreq, gwScrolledW2, i-Fit, iGramH, iGramW, iSpectH, iSpectW, and ReConfigDisplay().

Referenced by WrapOnSizeAllocate().

7.96.3.33 void clSpectGUI::GdkInput (gpointer, gint, GdkInputCondition)

Definition at line 544 of file GUISpect.cc.

References bConnected, bFreezed, _stSpectRes::bLinear, bReConfig, bRun, cp-GramXTime, cpRcvMsgBuf, DrawSpectrogram(), DrawSpectrum(), fGramX, fGramY, _stSpectRes::fPeakLevel, fpSpect, fSpectX, fSpectY, clSockOp::Get-Errno(), clSpectMsg::GetResult(), giGdkTag, guCtxtSB, gwEHighFreq, gw-ELowFreq, gwStatusBar, iClips, _stSpectRes::iHighFreq, _stSpectRes::iLow-Freq, iRcvMsgSize, _stSpectRes::lLength, PrintRealTime(), clSockOp::Read-N(), clSockOp::ReadSelect(), ReConfigDisplay(), SGUI_CONV_MAXLEN, SockOp, SpectMsg, and sSResult.

Referenced by WrapGdkInput().

7.96.3.34 void clSpectGUI::OnSaveClicks (GtkButton *, gpointer)

Definition at line 602 of file GUISpect.cc.

References FB_TIFF_COMPRESS_DEFLATE, FB_TIFF_COMPRESS_JPEG, FB_TIFF_COMPRESS_LZW, FB_TIFF_COMPRESS_NONE, FB_TIFF_-COMPRESS_RLE, FrameBuf, gwFSSave, iJPEGQuality, iTIFFCompression, SaveInfo(), and clFrameBuf::SaveToFile().

Referenced by WrapOnSaveClicks().

7.96.4 Member Data Documentation

7.96.4.1 bool clSpectGUI::bRun [private]

Definition at line 96 of file GUISpect.hh.

Referenced by clSpectGUI(), GdkInput(), OnDelete(), OnExposeSgram(), and OnExposeSpect().

7.96.4.2 bool clSpectGUI::bConnected [private]

Definition at line 97 of file GUISpect.hh.

Referenced by clSpectGUI(), GdkInput(), InitConnection(), OnConnectClick(), and ∼clSpectGUI().

7.96.4.3 bool clSpectGUI::bFreezed [private]

Definition at line 98 of file GUISpect.hh.

Referenced by clSpectGUI(), GdkInput(), and OnFreezeToggled().

7.96.4.4 bool clSpectGUI::bReConfig [private]

Definition at line 99 of file GUISpect.hh.

Referenced by GdkInput(), ReConfigDisplay(), and SendSettings().

7.96.4.5 int clSpectGUI::iSockH [private]

Definition at line 100 of file GUISpect.hh.

Referenced by InitConnection().

7.96.4.6 int clSpectGUI::iRcvMsgSize [private]

Definition at line 101 of file GUISpect.hh.

Referenced by GdkInput(), and GetSettings().

7.96.4.7 int clSpectGUI::iFit [private]

Definition at line 102 of file GUISpect.hh.

Referenced by Build(), clSpectGUI(), DrawSpectrogram(), OnConfigure(), OnSizeAllocate(), and ReConfigDisplay().

7.96.4.8 int clSpectGUI::iPalette [private]

Definition at line 103 of file GUISpect.hh.

Referenced by BuildTable3(), and clSpectGUI().

7.96.4.9 int clSpectGUI::iGramW [private]

Definition at line 104 of file GUISpect.hh.

Referenced by Build(), DrawSpectrogram(), OnConfigure(), OnExposeSgram(), OnMotionSgram(), OnSizeAllocate(), and ReConfigDisplay().

7.96.4.10 int clSpectGUI::iGramH [private]

Definition at line 105 of file GUISpect.hh.

Referenced by Build(), DrawSpectrogram(), OnConfigure(), OnExposeSgram(), OnMotionSgram(), OnSizeAllocate(), and ReConfigDisplay().

7.96.4.11 int clSpectGUI::iSpectW [private]

Definition at line 106 of file GUISpect.hh.

Referenced by OnConfigure(), and OnSizeAllocate().

7.96.4.12 int clSpectGUI::iSpectH [private]

Definition at line 107 of file GUISpect.hh.

Referenced by DrawSpectrum(), OnConfigure(), OnMotionSpect(), OnSizeAllocate(), and ReConfigDisplay().

7.96.4.13 int clSpectGUI::iScaleFactor [private]

Definition at line 108 of file GUISpect.hh.

7.96.4.14 int clSpectGUI::iSpectPoints [private]

Definition at line 109 of file GUISpect.hh.

Referenced by DrawSpectrogram(), DrawSpectrum(), OnExposeSpect(), OnMotionSpect(), and ReConfigDisplay().

7.96.4.15 int clSpectGUI::iClips [private]

Definition at line 110 of file GUISpect.hh.

Referenced by GdkInput(), InitConnection(), OnMotionSgram(), and OnMotionSpect().

7.96.4.16 int clSpectGUI::iTIFFCompression [private]

Definition at line 111 of file GUISpect.hh.

Referenced by clSpectGUI(), and OnSaveClicks().

7.96.4.17 int clSpectGUI::iJPEGQuality [private]

Definition at line 112 of file GUISpect.hh.

Referenced by clSpectGUI(), and OnSaveClicks().

7.96.4.18 int clSpectGUI::iBeamCount [private]

Definition at line 113 of file GUISpect.hh.

Referenced by BuildTable1(), clSpectGUI(), and GetSettings().

7.96.4.19 float clSpectGUI::fGramX [private]

Definition at line 114 of file GUISpect.hh.

Referenced by Build(), GdkInput(), OnMotionSgram(), OnMotionSpect(), and PrintRealTime().

7.96.4.20 float clSpectGUI::fGramY [private]

Definition at line 115 of file GUISpect.hh.

Referenced by Build(), GdkInput(), OnMotionSgram(), and OnMotionSpect().

7.96.4.21 float clSpectGUI::fSpectX [private]

Definition at line 116 of file GUISpect.hh.

Referenced by Build(), GdkInput(), OnMotionSgram(), and OnMotionSpect().

7.96.4.22 float clSpectGUI::fSpectY [private]

Definition at line 117 of file GUISpect.hh.

Referenced by Build(), GdkInput(), OnMotionSgram(), and OnMotionSpect().

7.96.4.23 char clSpectGUI::cpGramXTime[20] [private]

Definition at line 118 of file GUISpect.hh.

Referenced by GdkInput(), OnMotionSgram(), OnMotionSpect(), and PrintRealTime().

7.96.4.24 char* clSpectGUI::cpRcvMsgBuf [private]

Definition at line 119 of file GUISpect.hh.

Referenced by clSpectGUI(), GdkInput(), and GetSettings().

7.96.4.25 GDT* clSpectGUI::fpSpect [private]

Definition at line 120 of file GUISpect.hh.

Referenced by clSpectGUI(), DrawSpectrogram(), DrawSpectrum(), GdkInput(), GetSettings(), and OnExposeSpect().

7.96.4.26 GDT* clSpectGUI::fpIntSpect [private]

Definition at line 121 of file GUISpect.hh.

Referenced by DrawSpectrogram().

7.96.4.27 gint clSpectGUI::giGdkTag [private]

Definition at line 122 of file GUISpect.hh.

Referenced by GdkInput(), InitConnection(), and OnConnectClick().

7.96.4.28 GList* clSpectGUI::glServers [private]

Definition at line 123 of file GUISpect.hh.

Referenced by BuildTable1(), and clSpectGUI().

7.96.4.29 GtkWidget* clSpectGUI::gwWindow [private]

Definition at line 125 of file GUISpect.hh.

Referenced by Build(), and ConnectSignals().

7.96.4.30 GtkWidget* clSpectGUI::gwVBox [private]

Definition at line 126 of file GUISpect.hh.

Referenced by Build(), BuildTable1(), BuildTable2(), and BuildTable3().

7.96.4.31 GtkWidget* clSpectGUI::gwCBHide [private]

Definition at line 127 of file GUISpect.hh.

Referenced by Build(), and ConnectSignals().

7.96.4.32 GtkWidget* clSpectGUI::gwVPaned [private]

Definition at line 128 of file GUISpect.hh.

Referenced by Build().

7.96.4.33 GtkWidget* clSpectGUI::gwScrolledW1 [private]

Definition at line 129 of file GUISpect.hh.

Referenced by Build(), BuildTableGram(), DrawSpectrogram(), and GetGram-Height().

7.96.4.34 GtkWidget* clSpectGUI::gwScrolledW2 [private]

Definition at line 130 of file GUISpect.hh.

Referenced by Build(), BuildTableSpect(), ConnectSignals(), DrawSpectrum(), and OnSizeAllocate().

7.96.4.35 GtkWidget* clSpectGUI::gwStatusBar [private]

Definition at line 131 of file GUISpect.hh.

Referenced by Build(), GdkInput(), OnMotionSgram(), and OnMotionSpect().

7.96.4.36 guint clSpectGUI::guCtxtSB [private]

Definition at line 132 of file GUISpect.hh.

Referenced by Build(), GdkInput(), OnMotionSgram(), and OnMotionSpect().

7.96.4.37 GtkWidget* clSpectGUI::gwTable1 [private]

Definition at line 134 of file GUISpect.hh.

Referenced by BuildTable1(), and OnHideToggled().

7.96.4.38 GtkWidget* clSpectGUI::gwLServer [private]

Definition at line 135 of file GUISpect.hh.

Referenced by BuildTable1().

7.96.4.39 GtkWidget* clSpectGUI::gwCServer [private]

Definition at line 136 of file GUISpect.hh.

Referenced by BuildTable1(), and OnConnectClick().

7.96.4.40 GtkWidget* clSpectGUI::gwLChannel [private]

Definition at line 137 of file GUISpect.hh.

Referenced by BuildTable1().

7.96.4.41 GtkObject* clSpectGUI::goAChannel [private]

Definition at line 138 of file GUISpect.hh.

Referenced by BuildTable1().

7.96.4.42 GtkWidget* clSpectGUI::gwSBChannel [private]

Definition at line 139 of file GUISpect.hh.

Referenced by BuildTable1(), and GetSettings().

7.96.4.43 GtkWidget* clSpectGUI::gwBConnect [private]

Definition at line 140 of file GUISpect.hh.

Referenced by BuildTable1(), ConnectSignals(), and OnConnectClick().

7.96.4.44 GtkWidget* clSpectGUI::gwBDisconnect [private]

Definition at line 141 of file GUISpect.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.96.4.45 GtkWidget* clSpectGUI::gwCBFreeze [private]

Definition at line 142 of file GUISpect.hh.

Referenced by BuildTable1(), and ConnectSignals().

7.96.4.46 GtkWidget* clSpectGUI::gwTable2 [private]

Definition at line 144 of file GUISpect.hh.

Referenced by BuildTable2(), and OnHideToggled().

7.96.4.47 GtkWidget* clSpectGUI::gwLType [private]

Definition at line 145 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.48 GtkWidget* clSpectGUI::gwOMType [private]

Definition at line 146 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.49 GtkWidget* clSpectGUI::gwMType [private]

Definition at line 147 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.50 GtkWidget* clSpectGUI::gwaMIType[SGUI_TYPE_ITEMS] [private]

Definition at line 148 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.51 GtkWidget* clSpectGUI::gwLWindow [private]

Definition at line 149 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.52 GtkWidget* clSpectGUI::gwOMWindow [private]

Definition at line 150 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.53 GtkWidget* clSpectGUI::gwMWindow [private]

Definition at line 151 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.54 GtkWidget* clSpectGUI::gwaMIWindow[SGUI_WINDOW_ITEMS] [private]

Definition at line 152 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.55 GtkWidget* clSpectGUI::gwLWindowParam [private]

Definition at line 153 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.56 GtkWidget* clSpectGUI::gwEWindowParam [private]

Definition at line 154 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.57 GtkWidget* clSpectGUI::gwLWindowLen [private]

Definition at line 155 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.58 GtkWidget* clSpectGUI::gwOMWindowLen [private]

Definition at line 156 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.59 GtkWidget* clSpectGUI::gwMWindowLen [private]

Definition at line 157 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.60 GtkWidget* clSpectGUI::gwaMIWindowLen[SGUI_WINLEN_ITEMS] [private]

Definition at line 158 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.61 GtkWidget* clSpectGUI::gwLLowFreq [private]

Definition at line 159 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.62 GtkWidget* clSpectGUI::gwELowFreq [private]

Definition at line 160 of file GUISpect.hh.

Referenced by BuildTable2(), GdkInput(), and GetSettings().

7.96.4.63 GtkWidget* clSpectGUI::gwLHighFreq [private]

Definition at line 161 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.64 GtkWidget* clSpectGUI::gwEHighFreq [private]

Definition at line 162 of file GUISpect.hh.

Referenced by BuildTable2(), GdkInput(), and GetSettings().

7.96.4.65 GtkWidget* clSpectGUI::gwLGain [private]

Definition at line 163 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.66 GtkWidget* clSpectGUI::gwEGain [private]

Definition at line 164 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.67 GtkWidget* clSpectGUI::gwLSlope [private]

Definition at line 165 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.68 GtkWidget* clSpectGUI::gwESlope [private]

Definition at line 166 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.69 GtkWidget* clSpectGUI::gwLOverlap [private]

Definition at line 167 of file GUISpect.hh.

Referenced by BuildTable2().

7.96.4.70 GtkWidget* clSpectGUI::gwEOverlap [private]

Definition at line 168 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.71 GtkWidget* clSpectGUI::gwCBLinear [private]

Definition at line 169 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.72 GtkWidget* clSpectGUI::gwCBNormalize [private]

Definition at line 170 of file GUISpect.hh.

Referenced by BuildTable2(), and GetSettings().

7.96.4.73 GtkWidget* clSpectGUI::gwTable3 [private]

Definition at line 173 of file GUISpect.hh.

Referenced by BuildTable3(), and OnHideToggled().

7.96.4.74 GtkWidget* clSpectGUI::gwLRemoveNoise [private]

Definition at line 174 of file GUISpect.hh.

Referenced by BuildTable3().

7.96.4.75 GtkWidget* clSpectGUI::gwOMRemoveNoise [private]

Definition at line 175 of file GUISpect.hh.

Referenced by BuildTable3(), and GetSettings().

7.96.4.76 GtkWidget* clSpectGUI::gwMRemoveNoise [private]

Definition at line 176 of file GUISpect.hh.

Referenced by BuildTable3().

7.96.4.77 GtkWidget* clSpectGUI::gwaMIRemoveNoise[SGUI_REMOVE_NOISE_ITEMS] [private]

Definition at line 177 of file GUISpect.hh.

Referenced by BuildTable3(), and GetSettings().

7.96.4.78 GtkWidget* clSpectGUI::gwLAlpha [private]

Definition at line 178 of file GUISpect.hh.

Referenced by BuildTable3().

7.96.4.79 GtkWidget* clSpectGUI::gwEAlpha [private]

Definition at line 179 of file GUISpect.hh.

Referenced by BuildTable3(), and GetSettings().

7.96.4.80 GtkWidget* clSpectGUI::gwLMeanLength [private]

Definition at line 180 of file GUISpect.hh.

Referenced by BuildTable3().

7.96.4.81 GtkWidget* clSpectGUI::gwEMeanLength [private]

Definition at line 181 of file GUISpect.hh.

Referenced by BuildTable3(), and GetSettings().

7.96.4.82 GtkWidget* clSpectGUI::gwLGapLength [private]

Definition at line 182 of file GUISpect.hh.

Referenced by BuildTable3().

7.96.4.83 GtkWidget* clSpectGUI::gwEGapLength [private]

Definition at line 183 of file GUISpect.hh.

Referenced by BuildTable3(), and GetSettings().

7.96.4.84 GtkWidget* clSpectGUI::gwLDynRange [private]

Definition at line 184 of file GUISpect.hh.

Referenced by BuildTable3().

7.96.4.85 GtkWidget* clSpectGUI::gwEDynRange [private]

Definition at line 185 of file GUISpect.hh.

Referenced by BuildTable3(), and GetSettings().

7.96.4.86 GtkWidget* clSpectGUI::gwLPalette [private]

Definition at line 186 of file GUISpect.hh.

Referenced by BuildTable3().

7.96.4.87 GtkWidget* clSpectGUI::gwOMPalette [private]

Definition at line 187 of file GUISpect.hh.

Referenced by BuildTable3().

7.96.4.88 GtkWidget* clSpectGUI::gwMPalette [private]

Definition at line 188 of file GUISpect.hh.

Referenced by BuildTable3().

7.96.4.89 GtkWidget* clSpectGUI::gwaMIPalette[SGUI_PALETTE_ITEMS] [private]

Definition at line 189 of file GUISpect.hh.

Referenced by BuildTable3(), ConnectSignals(), and OnPaletteActivate().

7.96.4.90 GtkWidget* clSpectGUI::gwBSave [private]

Definition at line 190 of file GUISpect.hh.

Referenced by BuildTable3(), and ConnectSignals().

7.96.4.91 GtkWidget* clSpectGUI::gwFSSave [private]

Definition at line 191 of file GUISpect.hh.

Referenced by BuildTable3(), ConnectSignals(), and OnSaveClicks().

7.96.4.92 GtkWidget* clSpectGUI::gwTableGram [private]

Definition at line 193 of file GUISpect.hh.

Referenced by BuildTableGram().

7.96.4.93 GtkWidget* clSpectGUI::gwHRTime [private]

Definition at line 194 of file GUISpect.hh.

Referenced by BuildTableGram(), ConnectSignals(), DrawSpectrogram(), GetGramHeight(), and ReConfigDisplay().

7.96.4.94 GtkWidget* clSpectGUI::gwVRFreq [private]

Definition at line 195 of file GUISpect.hh.

Referenced by BuildTableGram(), ConnectSignals(), and ReConfigDisplay().

7.96.4.95 GtkWidget∗ clSpectGUI::gwDASpectrogram [private]

Definition at line 196 of file GUISpect.hh.

Referenced by Build(), BuildDrawingPrims(), BuildTableGram(), ConnectSignals(), DrawSpectrogram(), OnConfigure(), OnSizeAllocate(), and ReConfigDisplay().

7.96.4.96 GtkWidget∗ clSpectGUI::gwTableSpect [private]

Definition at line 198 of file GUISpect.hh.

Referenced by BuildTableSpect().

7.96.4.97 GtkWidget∗ clSpectGUI::gwHRFreq [private]

Definition at line 199 of file GUISpect.hh.

Referenced by BuildTableSpect(), ConnectSignals(), OnSizeAllocate(), and ReConfigDisplay().

7.96.4.98 GtkWidget∗ clSpectGUI::gwVRLevel [private]

Definition at line 200 of file GUISpect.hh.

Referenced by BuildTableSpect(), ConnectSignals(), and ReConfigDisplay().

7.96.4.99 GtkWidget∗ clSpectGUI::gwDASpectrum [private]

Definition at line 201 of file GUISpect.hh.

Referenced by Build(), BuildDrawingPrims(), BuildTableSpect(), ConnectSignals(), DrawSpectrum(), OnConfigure(), OnSizeAllocate(), and ReConfigDisplay().

7.96.4.100 GdkGC∗ clSpectGUI::ggcGramBG [private]

Definition at line 204 of file GUISpect.hh.

Referenced by BuildDrawingPrims(), and FreeDrawingPrims().

7.96.4.101 GdkGC∗ clSpectGUI::ggcGramFG [private]

Definition at line 205 of file GUISpect.hh.

Referenced by BuildDrawingPrims(), DrawSpectrogram(), FreeDrawingPrims(), and OnExposeSgram().

7.96.4.102 GdkGC* clSpectGUI::ggcSpectBG [private]

Definition at line 206 of file GUISpect.hh.

Referenced by BuildDrawingPrims(), and FreeDrawingPrims().

7.96.4.103 GdkGC* clSpectGUI::ggcSpectFG [private]

Definition at line 207 of file GUISpect.hh.

Referenced by BuildDrawingPrims(), DrawSpectrum(), FreeDrawingPrims(), and OnExposeSpect().

7.96.4.104 GdkCursor* clSpectGUI::gcCrossHair [private]

Definition at line 208 of file GUISpect.hh.

Referenced by Build(), BuildDrawingPrims(), and FreeDrawingPrims().

7.96.4.105 stSpectReq clSpectGUI::sSRequest [private]

Definition at line 210 of file GUISpect.hh.

Referenced by GetSettings(), OnMotionSpect(), ReConfigDisplay(), and Send-Settings().

7.96.4.106 stSpectRes clSpectGUI::sSResult [private]

Definition at line 211 of file GUISpect.hh.

Referenced by Build(), GdkInput(), OnMotionSgram(), OnMotionSpect(), PrintRealTime(), ReConfigDisplay(), and SaveInfo().

7.96.4.107 clAlloc clSpectGUI::RcvMsgBuf [private]

Definition at line 212 of file GUISpect.hh.

Referenced by GetSettings().

7.96.4.108 clAlloc clSpectGUI::SpectBuf [private]

Definition at line 213 of file GUISpect.hh.

Referenced by GetSettings().

7.96.4.109 clAlloc clSpectGUI::IntSpectBuf [private]

Definition at line 214 of file GUISpect.hh.

Referenced by DrawSpectrogram().

7.96.4.110 clCfgFile* clSpectGUI::CfgFile [private]

Definition at line 215 of file GUISpect.hh.

Referenced by Build(), BuildDrawingPrims(), clSpectGUI(), OnPaletteActivate(), and ~clSpectGUI().

7.96.4.111 clDSPOp clSpectGUI::DSP [private]

Definition at line 216 of file GUISpect.hh.

Referenced by DrawSpectrogram().

7.96.4.112 clGtkUtils clSpectGUI::GtkUtils [private]

Definition at line 217 of file GUISpect.hh.

Referenced by BuildTable1(), BuildTable2(), BuildTable3(), ConnectSignals(), and GetSettings().

7.96.4.113 clSockClie clSpectGUI::SockClie [private]

Definition at line 218 of file GUISpect.hh.

Referenced by InitConnection().

7.96.4.114 clSockOp* clSpectGUI::SockOp [private]

Definition at line 219 of file GUISpect.hh.

Referenced by GdkInput(), InitConnection(), OnConnectClick(), SendSettings(), and ~clSpectGUI().

7.96.4.115 clSpectMsg clSpectGUI::SpectMsg [private]

Definition at line 220 of file GUISpect.hh.

Referenced by GdkInput(), and SendSettings().

7.96.4.116 clFrameBuf clSpectGUI::FrameBuf [private]

Definition at line 221 of file GUISpect.hh.

Referenced by Build(), BuildDrawingPrims(), DrawSpectrogram(), OnExposeSgram(), OnPaletteActivate(), OnSaveClicks(), and ReConfigDisplay().

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

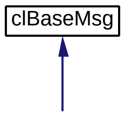
- **GUISpect.hh**
- **GUISpect.cc**

7.97 clSpectMsg Class Reference

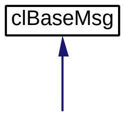
Spectrum server communication.

#include <Messages.hh>

Inheritance diagram for clSpectMsg:



Collaboration diagram for clSpectMsg:



Public Member Functions

- void **SetRequest** (char *, const **stpSpectReq**)
- void **GetRequest** (const char *, **stpSpectReq**)
- void **SetResult** (void *, const **stpSpectRes**, const float *)
- void **SetResult** (void *, const **stpSpectRes**, const double *)
- void **GetResult** (const void *, **stpSpectRes**, float *)
- void **GetResult** (const void *, **stpSpectRes**, double *)

7.97.1 Detailed Description

Spectrum server communication.

Definition at line 498 of file Messages.hh.

7.97.2 Member Function Documentation

7.97.2.1 void clSpectMsg::SetRequest (char *, const *stpSpectReq*)

Definition at line 243 of file Messages.cc.

References _stSpectReq::bLinear, _stSpectReq::bNormalize, _stSpectReq::f-Alpha, _stSpectReq::fDynRange, _stSpectReq::fGain, _stSpectReq::fSlope, _st-

SpectReq::fWinParam, GLOBAL_HEADER_LEN, _stSpectReq::iChannel, _stSpectReq::iHighFreq, _stSpectReq::iLowFreq, _stSpectReq::iOverlap, _stSpectReq::iRemoveNoise, _stSpectReq::iType, _stSpectReq::iWindow, _stSpectReq::lGapLength, _stSpectReq::lLength, _stSpectReq::lMeanLength, MSG_FALSE, and MSG_TRUE.

Referenced by clSpectGUI::SendSettings().

7.97.2.2 void clSpectMsg::GetRequest (const char *, stpSpectReq)

Definition at line 294 of file Messages.cc.

References _stSpectReq::bLinear, _stSpectReq::bNormalize, _stSpectReq::fAlpha, _stSpectReq::fDynRange, _stSpectReq::fGain, _stSpectReq::fSlope, _stSpectReq::fWinParam, _stSpectReq::iChannel, _stSpectReq::iHighFreq, _stSpectReq::iLowFreq, _stSpectReq::iOverlap, _stSpectReq::iRemoveNoise, _stSpectReq::iType, _stSpectReq::iWindow, _stSpectReq::lGapLength, _stSpectReq::lLength, _stSpectReq::lMeanLength, and MSG_TRUE.

Referenced by clSpectrum::GetRq().

7.97.2.3 void clSpectMsg::SetResult (void *, const *stpSpectRes*, const float *)

Definition at line 322 of file Messages.cc.

References _stSpectRes::bLinear, clBaseMsg::EndianConv(), _stSpectRes::fLineTime, _stSpectRes::fPeakLevel, GLOBAL_HEADER_LEN, _stSpectRes::iChannel, _stSpectRes::iHighFreq, _stSpectRes::iLowFreq, _stSpectRes::iSampleRate, _stSpectRes::lLength, MSG_FALSE, MSG_TRUE, and _stSpectRes::sTimeStamp.

Referenced by clSpectrum::ProcessLoop().

7.97.2.4 void clSpectMsg::SetResult (void *, const *stpSpectRes*, const double *)

Definition at line 364 of file Messages.cc.

References _stSpectRes::bLinear, clBaseMsg::EndianConv(), _stSpectRes::fLineTime, _stSpectRes::fPeakLevel, GLOBAL_HEADER_LEN, _stSpectRes::iChannel, _stSpectRes::iHighFreq, _stSpectRes::iLowFreq, _stSpectRes::iSampleRate, _stSpectRes::lLength, MSG_FALSE, MSG_TRUE, and _stSpectRes::sTimeStamp.

7.97.2.5 void clSpectMsg::GetResult (const void *, stpSpectRes, float *)

Definition at line 406 of file Messages.cc.

References _stSpectRes::bLinear, clBaseMsg::EndianConv(), _stSpectRes::fLineTime, _stSpectRes::fPeakLevel, GLOBAL_HEADER_LEN, _stSpectRes::iChannel, _stSpectRes::iHighFreq, _stSpectRes::iLowFreq, _stSpectRes::iSampleRate, _stSpectRes::lLength, MSG_TRUE, and _stSpectRes::sTimeStamp.

Referenced by clSpectGUI::GdkInput().

7.97.2.6 void clSpectMsg::GetResult (const void *, stpSpectRes, double *)

Definition at line 436 of file Messages.cc.

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

- **Messages.hh**
- **Messages.cc**

7.98 _stSpectReq Struct Reference

Spectrum: request.

`#include <Messages.hh>`

Public Attributes

- int **iChannel**
  Channel.
- int **iType**
  Type (STFT, Hankel, etc).
- int **iWindow**
  Window function.
- float **fWinParam**
  Optional window parameter.
- long **lLength**
  Number of points.
- int **iLowFreq**
  Lower frequency limit (Hz).
- int **iHighFreq**
  Higher frequency limit (Hz).
- float **fGain**
  Gain (dB).
- float **fSlope**
  Gain (dB/oct).
- int **iOverlap**
  Overlap ().
- bool **bLinear**
  Linear/logarithmic scaling.
- bool **bNormalize**
  Normalize?
- int **iRemoveNoise**
  Noise removal algorithm.

- float **fAlpha**

  Alpha for noise removal.

- long **lMeanLength**

  Mean length for noise removal.

- long **lGapLength**

  Gap length for TPSW noise removal.

- float **fDynRange**

  Dynamic range for logarithmic (dB) level.

7.98.1 Detailed Description

Spectrum: request.

Definition at line 210 of file Messages.hh.

7.98.2 Member Data Documentation

7.98.2.1 int _stSpectReq::iChannel

Channel.

Definition at line 212 of file Messages.hh.

Referenced by clSpectrum::ConnectStream(), clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.2 int _stSpectReq::iType

Type (STFT, Hankel, etc).

Definition at line 213 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::Init(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.3 int _stSpectReq::iWindow

Window function.

Definition at line 214 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::Init(), and clSpectMsg::SetRequest().

7.98.2.4 float _stSpectReq::fWinParam

Optional window parameter.

Definition at line 215 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::Init(), and clSpectMsg::SetRequest().

7.98.2.5 long _stSpectReq::lLength

Number of points.

Definition at line 216 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::Init(), and clSpectMsg::SetRequest().

7.98.2.6 int _stSpectReq::iLowFreq

Lower frequency limit (Hz).

Definition at line 217 of file Messages.hh.

Referenced by clSpectrum::CreateFilter(), clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), and clSpectMsg::SetRequest().

7.98.2.7 int _stSpectReq::iHighFreq

Higher frequency limit (Hz).

Definition at line 218 of file Messages.hh.

Referenced by clSpectrum::CreateFilter(), clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), and clSpectMsg::SetRequest().

7.98.2.8 float _stSpectReq::fGain

Gain (dB).

Definition at line 219 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.9 float _stSpectReq::fSlope

Gain (dB/oct).

Definition at line 220 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.10 int _stSpectReq::iOverlap

Overlap ().

Definition at line 221 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::Init(), and clSpectMsg::SetRequest().

7.98.2.11 bool _stSpectReq::bLinear

Linear/logarithmic scaling.

Definition at line 222 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.12 bool _stSpectReq::bNormalize

Normalize?

Definition at line 223 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.13 int _stSpectReq::iRemoveNoise

Noise removal algorithm.

Definition at line 224 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.14 float _stSpectReq::fAlpha

Alpha for noise removal.

Definition at line 225 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.15 long _stSpectReq::lMeanLength

Mean length for noise removal.

Definition at line 226 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.16 long _stSpectReq::lGapLength

Gap length for TPSW noise removal.

Definition at line 227 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectrum::ProcessLoop(), and clSpectMsg::SetRequest().

7.98.2.17 float _stSpectReq::fDynRange

Dynamic range for logarithmic (dB) level.

Definition at line 228 of file Messages.hh.

Referenced by clSpectMsg::GetRequest(), clSpectGUI::GetSettings(), clSpectGUI::OnMotionSpect(), clSpectrum::ProcessLoop(), clSpectGUI::ReConfigDisplay(), and clSpectMsg::SetRequest().

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

- **Messages.hh**

7.99 _stSpectRes Struct Reference

Spectrum: result.

`#include <Messages.hh>`

Public Attributes

- timeval **sTimeStamp**
  Timestamp.
- int **iChannel**
  Channel.
- long **lLength**
  Result length.
- int **iLowFreq**
  Lower frequency limit (may differ from request).
- int **iHighFreq**
  Higher frequency limit (may differ from request).
- int **iSampleRate**
  Samplerate.
- bool **bLinear**
  Linear/logarithmic.
- float **fPeakLevel**
  Peak level (dB).
- float **fLineTime**
  Length of spectrum window (s).

7.99.1 Detailed Description

Spectrum: result.

Definition at line 234 of file Messages.hh.

7.99.2 Member Data Documentation

7.99.2.1 struct timeval _stSpectRes::sTimeStamp

Timestamp.

Definition at line 236 of file Messages.hh.

Referenced by clSpectMsg::GetResult(), clSpectGUI::PrintRealTime(), clSpectrum::ProcessLoop(), and clSpectMsg::SetResult().

7.99.2.2 int _stSpectRes::iChannel

Channel.

Definition at line 237 of file Messages.hh.

Referenced by clSpectMsg::GetResult(), clSpectrum::ProcessLoop(), and clSpectMsg::SetResult().

7.99.2.3 long _stSpectRes::lLength

Result length.

Definition at line 238 of file Messages.hh.

Referenced by clSpectGUI::Build(), clSpectGUI::GdkInput(), clSpectMsg::GetResult(), clSpectrum::ProcessLoop(), clSpectGUI::ReConfigDisplay(), and clSpectMsg::SetResult().

7.99.2.4 int _stSpectRes::iLowFreq

Lower frequency limit (may differ from request).

Definition at line 239 of file Messages.hh.

Referenced by clSpectGUI::GdkInput(), clSpectMsg::GetResult(), clSpectGUI::OnMotionSgram(), clSpectGUI::OnMotionSpect(), clSpectrum::ProcessLoop(), clSpectGUI::ReConfigDisplay(), clSpectGUI::SaveInfo(), and clSpectMsg::SetResult().

7.99.2.5 int _stSpectRes::iHighFreq

Higher frequency limit (may differ from request).

Definition at line 240 of file Messages.hh.

Referenced by clSpectGUI::GdkInput(), clSpectMsg::GetResult(), clSpectGUI::OnMotionSgram(), clSpectGUI::OnMotionSpect(), clSpectrum::ProcessLoop(), clSpectGUI::ReConfigDisplay(), clSpectGUI::SaveInfo(), and clSpectMsg::SetResult().

7.99.2.6 int _stSpectRes::iSampleRate

Samplerate.

Definition at line 241 of file Messages.hh.

Referenced by clSpectMsg::GetResult(), clSpectrum::ProcessLoop(), and clSpectMsg::SetResult().

7.99.2.7 bool _stSpectRes::bLinear

Linear/logarithmic.

Definition at line 242 of file Messages.hh.

Referenced by clSpectGUI::GdkInput(), clSpectMsg::GetResult(), clSpectrum::ProcessLoop(), clSpectGUI::ReConfigDisplay(), and clSpectMsg::SetResult().

7.99.2.8 float _stSpectRes::fPeakLevel

Peak level (dB).

Definition at line 243 of file Messages.hh.

Referenced by clSpectGUI::Build(), clSpectGUI::GdkInput(), clSpectMsg::GetResult(), clSpectGUI::OnMotionSgram(), clSpectGUI::OnMotionSpect(), clSpectrum::ProcessLoop(), and clSpectMsg::SetResult().

7.99.2.9 float _stSpectRes::fLineTime

Length of spectrum window (s).

Definition at line 244 of file Messages.hh.

Referenced by clSpectMsg::GetResult(), clSpectGUI::OnMotionSgram(), clSpectrum::ProcessLoop(), clSpectGUI::ReConfigDisplay(), clSpectGUI::SaveInfo(), and clSpectMsg::SetResult().

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

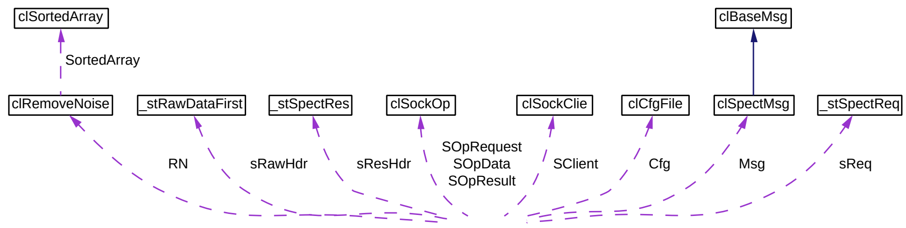
- **Messages.hh**

7.100 clSpectrum Class Reference

Spectrum server.

`#include <Spectrum.hh>`

Collaboration diagram for clSpectrum:



Public Member Functions

- **clSpectrum** (int, int)
- **~clSpectrum** ()
- int **Main** ()
- void **Stop** ()

Private Member Functions

- bool **GetCfg** ()
- bool **GetRq** ()
- bool **ConnectStream** ()
- bool **Init** ()
- void **CreateFilter** ()
- void **ProcessLoop** ()
- bool **GetData** ()
- bool **PullData** (GDT ∗, long)

Private Attributes

- bool **bRun**
- bool **bReverseOrder**
- int **iFilterType**
- int **iInDataSize**
- long **lInDataCount**
- long **lFilterSize**
- long **lDecimation**
- long **lOldDataCount**
- long **lNewDataCount**

- long **lSpectPoints**
- long **lSpectLen**
- long **lResSize**
- float **fLowCorner**
- float **fHighCorner**
- char **cpStreamSocket** [_POSIX_PATH_MAX+1]
- **stSpectReq sReq**
- **stSpectRes sResHdr**
- **stRawDataFirst sRawHdr**
- clDSPAlloc **InData**
- clDSPAlloc **Window**
- clDSPAlloc **SpectIn**
- clDSPAlloc **SpectOut**
- clDSPAlloc **ResMsg**
- **clCfgFile Cfg**
- clDSPOp **DSP**
- clDSPOp **FFT**
- clHankel **Hankel**
- clReBuffer **ReBuffer**
- clRecDecimator **Decimator**
- clRecDecimator **WVDDec**
- **clRemoveNoise RN**
- **clSpectMsg Msg**
- **clSockClie SClient**
- **clSockOp SOpRequest**
- **clSockOp SOpResult**
- **clSockOp SOpData**

7.100.1 Detailed Description

Spectrum server.

Definition at line 46 of file Spectrum.hh.

7.100.2 Constructor & Destructor Documentation

7.100.2.1 clSpectrum::clSpectrum (int, int)

Definition at line 562 of file Spectrum.cc.

References bRun, clSockOp::SetHandle(), SOpRequest, and SOpResult.

7.100.2.2 clSpectrum::~clSpectrum ()

Definition at line 570 of file Spectrum.cc.

7.100.3 Member Function Documentation

7.100.3.1 bool clSpectrum::GetCfg () `[private]`

Definition at line 102 of file Spectrum.cc.

References Cfg, cpStreamSocket, clCfgFile::GetInt(), clCfgFile::GetStr(), iFilterType, lFilterSize, SD_CFGFILE, clCfgFile::SetFileName(), SPECT_CFGFILE, and SPECT_DEF_FILTSIZE.

Referenced by Main().

7.100.3.2 bool clSpectrum::GetRq () `[private]`

Definition at line 115 of file Spectrum.cc.

References clSpectMsg::GetRequest(), GLOBAL_HEADER_LEN, Msg, clSockOp::ReadN(), clSockOp::ReadSelect(), SOpRequest, SPECT_REQ_TIMEOUT, and sReq.

Referenced by Main().

7.100.3.3 bool clSpectrum::ConnectStream () `[private]`

Definition at line 127 of file Spectrum.cc.

References clSockClie::Connect(), cpStreamSocket, _stSpectReq::iChannel, _stRawDataReq::iChannel, clSockOp::ReadN(), clSockOp::ReadSelect(), SClient, clSockOp::SetHandle(), SOpData, SPECT_RAW1ST_TIMEOUT, sRawHdr, sReq, and clSockOp::WriteN().

Referenced by Main().

7.100.3.4 bool clSpectrum::Init () `[private]`

Definition at line 145 of file Spectrum.cc.

References bDebug, cpaWindowFuncs, CreateFilter(), DSP, FFT, _stSpectReq::fWinParam, GDT, GLOBAL_HEADER_LEN, Hankel, iFilterType, iInDataSize, InData, _stSpectReq::iOverlap, _stSpectReq::iType, _stSpectReq::iWindow, lFilterSize, lInDataCount, _stSpectReq::lLength, lNewDataCount, lOldDataCount, lResSize, lSpectLen, lSpectPoints, MSG_SPECT_TYPE_AUTOCORR, MSG_SPECT_TYPE_CEPSTRUM, MSG_SPECT_TYPE_FFT, MSG_SPECT_TYPE_GABOR, MSG_SPECT_TYPE_HANKEL, MSG_SPECT_TYPE_MRFFT, MSG_SPECT_TYPE_WVD, MSG_SPECT_WND_BARTLETT, MSG_SPECT_WND_BLACKMAN, MSG_SPECT_WND_BM_HAR, MSG_SPECT_WND_COS_TAPER, MSG_SPECT_WND_EXP, MSG_SPECT_WND_FLATTOP, MSG_SPECT_WND_GEN_COS, MSG_SPECT_WND_HAMMING, MSG_SPECT_WND_HANNING, MSG_SPECT_WND_KAI_BES, MSG_SPECT_WND_KAISER, MSG_SPECT_WND_RECT, MSG_SPECT_WND_TUKEY, ResMsg, SPECT_NWIN, SpectIn, SpectOut, sReq, Window, and WVDDec.

Referenced by Main().

7.100.3.5 void clSpectrum::CreateFilter () [private]

Definition at line 281 of file Spectrum.cc.

References bDebug, bReverseOrder, Decimator, _stRawDataFirst::dSampleRate, fHighCorner, fLowCorner, GDT, iFilterType, _stSpectReq::iHighFreq, _stSpectReq::iLowFreq, lDecimation, lFilterSize, sRawHdr, and sReq.

Referenced by Init().

7.100.3.6 void clSpectrum::ProcessLoop () [private]

Definition at line 321 of file Spectrum.cc.

References bDebug, _stSpectRes::bLinear, _stSpectReq::bLinear, _stSpectReq::bNormalize, bReverseOrder, bRun, clRemoveNoise::Diff(), _stRawDataFirst::dSampleRate, DSP, _stSpectReq::fAlpha, _stSpectReq::fDynRange, FFT, _stSpectReq::fGain, fHighCorner, _stSpectRes::fLineTime, fLowCorner, _stSpectRes::fPeakLevel, _stSpectReq::fSlope, GCDT, GDT, GetData(), Hankel, _stSpectReq::iChannel, _stSpectRes::iChannel, _stSpectRes::iHighFreq, _stSpectRes::iLowFreq, clRemoveNoise::InvDiff(), _stSpectReq::iRemoveNoise, _stSpectRes::iSampleRate, _stSpectReq::iType, lDecimation, _stSpectReq::lGapLength, _stSpectRes::lLength, _stSpectReq::lMeanLength, lNewDataCount, lOldDataCount, lResSize, lSpectLen, lSpectPoints, Msg, MSG_SPECT_BNER_DIFF, MSG_SPECT_BNER_IDIFF, MSG_SPECT_BNER_NONE, MSG_SPECT_BNER_OTA, MSG_SPECT_BNER_TPSW, MSG_SPECT_TYPE_AUTOCORR, MSG_SPECT_TYPE_CEPSTRUM, MSG_SPECT_TYPE_FFT, MSG_SPECT_TYPE_GABOR, MSG_SPECT_TYPE_HANKEL, MSG_SPECT_TYPE_MRFFT, MSG_SPECT_TYPE_WVD, clRemoveNoise::OTA(), PullData(), ResMsg, RN, clSpectMsg::SetResult(), SOpResult, SpectIn, SpectOut, sRawHdr, sReq, sResHdr, _stSpectRes::sTimeStamp, Stop(), clRemoveNoise::TPSW(), Window, and clSockOp::WriteN().

Referenced by Main().

7.100.3.7 bool clSpectrum::GetData () [private]

Definition at line 535 of file Spectrum.cc.

References bDebug, Decimator, GDT, iInDataSize, InData, lDecimation, lInDataCount, clSockOp::ReadN(), ReBuffer, and SOpData.

Referenced by ProcessLoop().

7.100.3.8 bool clSpectrum::PullData (GDT *, long) [private]

Definition at line 553 of file Spectrum.cc.

References Decimator, lDecimation, and ReBuffer.

Referenced by ProcessLoop().

7.100.3.9 int clSpectrum::Main ()

Definition at line 575 of file Spectrum.cc.

References bDebug, ConnectStream(), GetCfg(), GetRq(), Init(), and ProcessLoop().

Referenced by main().

7.100.3.10 void clSpectrum::Stop () [inline]

Definition at line 96 of file Spectrum.hh.

References bRun.

Referenced by ProcessLoop().

7.100.4 Member Data Documentation

7.100.4.1 bool clSpectrum::bRun [private]

Definition at line 48 of file Spectrum.hh.

Referenced by clSpectrum(), ProcessLoop(), and Stop().

7.100.4.2 bool clSpectrum::bReverseOrder [private]

Definition at line 49 of file Spectrum.hh.

Referenced by CreateFilter(), and ProcessLoop().

7.100.4.3 int clSpectrum::iFilterType [private]

Definition at line 50 of file Spectrum.hh.

Referenced by CreateFilter(), GetCfg(), and Init().

7.100.4.4 int clSpectrum::iInDataSize [private]

Definition at line 51 of file Spectrum.hh.

Referenced by GetData(), and Init().

7.100.4.5 long clSpectrum::lInDataCount [private]

Definition at line 52 of file Spectrum.hh.

Referenced by GetData(), and Init().

7.100.4.6 long clSpectrum::lFilterSize [private]

Definition at line 53 of file Spectrum.hh.

Referenced by CreateFilter(), GetCfg(), and Init().

7.100.4.7 long clSpectrum::lDecimation [private]

Definition at line 54 of file Spectrum.hh.

Referenced by CreateFilter(), GetData(), ProcessLoop(), and PullData().

7.100.4.8 long clSpectrum::lOldDataCount [private]

Definition at line 55 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.9 long clSpectrum::lNewDataCount [private]

Definition at line 56 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.10 long clSpectrum::lSpectPoints [private]

Definition at line 57 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.11 long clSpectrum::lSpectLen [private]

Definition at line 58 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.12 long clSpectrum::lResSize [private]

Definition at line 59 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.13 float clSpectrum::fLowCorner [private]

Definition at line 60 of file Spectrum.hh.

Referenced by CreateFilter(), and ProcessLoop().

7.100.4.14 float clSpectrum::fHighCorner [private]

Definition at line 61 of file Spectrum.hh.

Referenced by CreateFilter(), and ProcessLoop().

7.100.4.15 char clSpectrum::cpStreamSocket[_POSIX_PATH_MAX + 1] [private]

Definition at line 62 of file Spectrum.hh.

Referenced by ConnectStream(), and GetCfg().

7.100.4.16 stSpectReq clSpectrum::sReq [private]

Definition at line 63 of file Spectrum.hh.

Referenced by ConnectStream(), CreateFilter(), GetRq(), Init(), and ProcessLoop().

7.100.4.17 stSpectRes clSpectrum::sResHdr [private]

Definition at line 64 of file Spectrum.hh.

Referenced by ProcessLoop().

7.100.4.18 stRawDataFirst clSpectrum::sRawHdr [private]

Definition at line 65 of file Spectrum.hh.

Referenced by ConnectStream(), CreateFilter(), and ProcessLoop().

7.100.4.19 clDSPAlloc clSpectrum::InData [private]

Definition at line 66 of file Spectrum.hh.

Referenced by GetData(), and Init().

7.100.4.20 clDSPAlloc clSpectrum::Window [private]

Definition at line 67 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.21 clDSPAlloc clSpectrum::SpectIn [private]

Definition at line 68 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.22 clDSPAlloc clSpectrum::SpectOut [private]

Definition at line 69 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.23 clDSPAlloc clSpectrum::ResMsg [private]

Definition at line 70 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.24 clCfgFile clSpectrum::Cfg [private]

Definition at line 71 of file Spectrum.hh.

Referenced by GetCfg().

7.100.4.25 clDSPOp clSpectrum::DSP [private]

Definition at line 72 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.26 clDSPOp clSpectrum::FFT [private]

Definition at line 73 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.27 clHankel clSpectrum::Hankel [private]

Definition at line 74 of file Spectrum.hh.

Referenced by Init(), and ProcessLoop().

7.100.4.28 clReBuffer clSpectrum::ReBuffer [private]

Definition at line 75 of file Spectrum.hh.

Referenced by GetData(), and PullData().

7.100.4.29 clRecDecimator clSpectrum::Decimator [private]

Definition at line 76 of file Spectrum.hh.

Referenced by CreateFilter(), GetData(), and PullData().

7.100.4.30 clRecDecimator clSpectrum::WVDDec [private]

Definition at line 77 of file Spectrum.hh.

Referenced by Init().

7.100.4.31 clRemoveNoise clSpectrum::RN [private]

Definition at line 78 of file Spectrum.hh.

Referenced by ProcessLoop().

7.100.4.32 clSpectMsg clSpectrum::Msg [private]

Definition at line 79 of file Spectrum.hh.

Referenced by GetRq(), and ProcessLoop().

7.100.4.33 clSockClie clSpectrum::SClient [private]

Definition at line 80 of file Spectrum.hh.

Referenced by ConnectStream().

7.100.4.34 clSockOp clSpectrum::SOpRequest [private]

Definition at line 81 of file Spectrum.hh.

Referenced by clSpectrum(), and GetRq().

7.100.4.35 clSockOp clSpectrum::SOpResult [private]

Definition at line 82 of file Spectrum.hh.

Referenced by clSpectrum(), and ProcessLoop().

7.100.4.36 clSockOp clSpectrum::SOpData [private]

Definition at line 83 of file Spectrum.hh.

Referenced by ConnectStream(), and GetData().

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

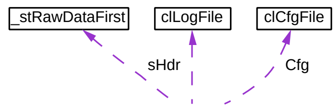
- **Spectrum.hh**
- **Spectrum.cc**

7.101 clStreamDist Class Reference

Audio stream distributor.

`#include <StreamDist.hh>`

Collaboration diagram for clStreamDist:



Public Member Functions

- **clStreamDist** ()
- ~**clStreamDist** ()
- int **Main** (int, char ∗∗)
- void ∗ **AudioInThread** (void ∗)
- void ∗ **ServeClientThread** (void ∗)
- void **Stop** ()

Public Attributes

- **clLogFile Log**

Private Member Functions

- void **CopyChannel** (GDT ∗, const GDT ∗, int)
- bool **InitCompress** (int)

Private Attributes

- volatile bool **bRun**
- volatile int **iFragmentSize**
- volatile int **iAudioBufSize**
- volatile int **iBlockCntr**
- **stRawDataFirst sHdr**
- **clCfgFile Cfg**
- clAlloc **AudioBuf**
- clMutex **MtxAudio**
- clCondition **CndAudio**

7.101.1 Detailed Description

Audio stream distributor.

Note:
See soundsrv2 for architecture description.

Definition at line 55 of file StreamDist.hh.

7.101.2 Constructor & Destructor Documentation

7.101.2.1 clStreamDist::clStreamDist ()

Definition at line 182 of file StreamDist.cc.

References clLogFile::Add(), bRun, Cfg, iAudioBufSize, iBlockCntr, Log, clLogFile::Open(), SD_CFGFILE, SD_LOGFILE, and clCfgFile::SetFileName().

7.101.2.2 clStreamDist::~clStreamDist ()

Definition at line 196 of file StreamDist.cc.

References clLogFile::Add(), and Log.

7.101.3 Member Function Documentation

7.101.3.1 void clStreamDist::CopyChannel (GDT *, const GDT *, int) `[inline, private]`

Definition at line 127 of file StreamDist.cc.

References _stRawDataFirst::iChannels, iFragmentSize, and sHdr.

Referenced by ServeClientThread().

7.101.3.2 bool clStreamDist::InitCompress (int) `[private]`

Definition at line 141 of file StreamDist.cc.

References clLogFile::Add(), Log, and MSG_SOUND_COMPRESS_FLAC.

Referenced by AudioInThread().

7.101.3.3 int clStreamDist::Main (int, char **)

Definition at line 208 of file StreamDist.cc.

References clLogFile::Add(), bDaemon, clSockServ::Bind(), bRun, Cfg, clSockServ::GetErrno(), clCfgFile::GetStr(), GLOBAL_VERSMAJ, GLOBAL_VERSMIN, GLOBAL_VERSPL, Log, SD_CONNECT_TIMEOUT, SD_SHUTDOWNFILE, SigHandler(), Stop(), clSockServ::WaitForConnect(), WrapAudioInThread(), and WrapServeClientThread().

Referenced by main().

7.101.3.4 void * clStreamDist::AudioInThread (void *)

Definition at line 288 of file StreamDist.cc.

References clLogFile::Add(), AudioBuf, bRun, Cfg, CndAudio, clSockClie::Connect(), _stRawDataFirst::dSampleRate, _stSoundStart::dSampleRate, GDT, clSoundMsg::GetData(), clSockOp::GetErrno(), clSockClie::GetErrno(), clCfgFile::GetInt(), clSoundMsg::GetStart(), clCfgFile::GetStr(), GLOBAL_HEADER_LEN, iAudioBufSize, iBlockCntr, _stRawDataFirst::iChannels, _stSoundStart::iChannels, _stSoundStart::iCompress, iFragmentSize, _stSoundStart::iFragmentSize, InitCompress(), Log, MSG_SOUND_COMPRESS_FLAC, MtxAudio, clSockOp::ReadN(), SD_HOSTADDR_MAX, SD_INTHREAD_PRIORITY, SD_LOGENTRY_SIZE, clSockOp::SetHandle(), sHdr, and SOp.

Referenced by WrapAudioInThread().

7.101.3.5 void * clStreamDist::ServeClientThread (void *)

Definition at line 415 of file StreamDist.cc.

References clLogFile::Add(), AudioBuf, bRun, CndAudio, CopyChannel(), iAudioBufSize, iBlockCntr, _stRawDataReq::iChannel, _stRawDataFirst::iChannels, Log, MtxAudio, clSockOp::ReadN(), SD_OUTTHREAD_PRIORITY, sHdr, SOp, and clSockOp::WriteN().

Referenced by WrapServeClientThread().

7.101.3.6 void clStreamDist::Stop () `[inline]`

Definition at line 84 of file StreamDist.hh.

References bRun.

Referenced by Main(), and SigHandler().

7.101.4 Member Data Documentation

7.101.4.1 volatile bool clStreamDist::bRun `[private]`

Definition at line 57 of file StreamDist.hh.

Referenced by AudioInThread(), clStreamDist(), Main(), ServeClientThread(), and Stop().

7.101.4.2 volatile int clStreamDist::iFragmentSize [private]

Definition at line 58 of file StreamDist.hh.

Referenced by AudioInThread(), and CopyChannel().

7.101.4.3 volatile int clStreamDist::iAudioBufSize [private]

Definition at line 59 of file StreamDist.hh.

Referenced by AudioInThread(), clStreamDist(), and ServeClientThread().

7.101.4.4 volatile int clStreamDist::iBlockCntr [private]

Definition at line 60 of file StreamDist.hh.

Referenced by AudioInThread(), clStreamDist(), and ServeClientThread().

7.101.4.5 stRawDataFirst clStreamDist::sHdr [private]

Definition at line 61 of file StreamDist.hh.

Referenced by AudioInThread(), CopyChannel(), and ServeClientThread().

7.101.4.6 clCfgFile clStreamDist::Cfg [private]

Definition at line 65 of file StreamDist.hh.

Referenced by AudioInThread(), clStreamDist(), and Main().

7.101.4.7 clAlloc clStreamDist::AudioBuf [private]

Definition at line 66 of file StreamDist.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.101.4.8 clMutex clStreamDist::MtxAudio [private]

Definition at line 67 of file StreamDist.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.101.4.9 clCondition clStreamDist::CndAudio [private]

Definition at line 74 of file StreamDist.hh.

Referenced by AudioInThread(), and ServeClientThread().

7.101.4.10 clLogFile clStreamDist::Log

Definition at line 78 of file StreamDist.hh.

Referenced by AudioInThread(), clStreamDist(), InitCompress(), Main(), ServeClientThread(), SigHandler(), and ~clStreamDist().

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

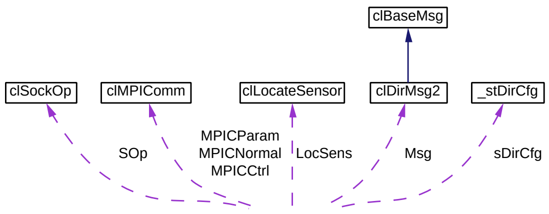
- **StreamDist.hh**
- **StreamDist.cc**

7.102 clSubLocate Class Reference

Slave nodes for locate server.

`#include <Locate.hh>`

Collaboration diagram for clSubLocate:



Public Member Functions

- **clSubLocate** ()
- ∼**clSubLocate** ()
- int **Main** (int, char ∗∗)

Private Member Functions

- bool **RecvParams** ()
- bool **Initialize** ()
- bool **ConnectDir** ()

Private Attributes

- int **iHostPort**
- int **iDirMsgSize**
- long **lWidth**
- long **lHeight**
- long **lPosX**
- long **lPosY**
- float **fAzimuth**
- char **cpHostName** [LOCATE_HOSTNAME_MAXLEN]
- GDT ∗ **fpLevRes**
- GDT ∗ **fpDirRes**
- **stDirCfg sDirCfg**
- clAlloc **DirMsg**
- clAlloc **LevRes**
- clAlloc **DirRes**
- **clLocateSensor LocSens**

- **clMPIComm MPICCtrl**
- **clMPIComm MPICParam**
- **clMPIComm MPICNormal**
- **clDirMsg2 Msg**
- **clSockOp SOp**

7.102.1 Detailed Description

Slave nodes for locate server.

Definition at line 110 of file Locate.hh.

7.102.2 Constructor & Destructor Documentation

7.102.2.1 clSubLocate::clSubLocate ()

Definition at line 420 of file Locate.cc.

References LOCATE_CONTROL_TAG, LOCATE_NORMAL_TAG, LOCATE_-PARAM_TAG, MPICCtrl, MPICNormal, MPICParam, and clMPIComm::Set-Tag().

7.102.2.2 clSubLocate::~clSubLocate ()

Definition at line 428 of file Locate.cc.

7.102.3 Member Function Documentation

7.102.3.1 bool clSubLocate::RecvParams () `[private]`

Definition at line 356 of file Locate.cc.

References cpHostName, fAzimuth, clMPIComm::GetSenderTag(), iHostPort, lHeight, LOCATE_HOSTNAME_MAXLEN, LOCATE_PARAM_TAG, lPosX, l-PosY, lWidth, MPICParam, clMPIComm::ProbeAny(), clMPIComm::Recv(), and sDirCfg.

Referenced by Main().

7.102.3.2 bool clSubLocate::Initialize () `[private]`

Definition at line 376 of file Locate.cc.

References DirMsg, DirRes, fAzimuth, fpDirRes, fpLevRes, GDT, GLOBAL_-HEADER_LEN, iDirMsgSize, clLocateSensor::Initialize(), LevRes, lHeight, LocSens, lPosX, lPosY, lWidth, _stDirCfg::lWindowSize, and sDirCfg.

Referenced by Main().

7.102.3.3 bool clSubLocate::ConnectDir () [private]

Definition at line 390 of file Locate.cc.

References clSockClie::Connect(), cpHostName, _stDirCfg::fAlpha, _stDirReq2::fAlpha, _stDirCfg::fIntegrationTime, _stDirReq2::fIntegrationTime, _stDirReq2::fLowFreqLimit, _stDirCfg::fLowFrequency, _stDirCfg::fScalingExp, _stDirReq2::fScalingExp, _stDirCfg::fSoundSpeed, _stDirReq2::fSoundSpeed, GLOBAL_HEADER_LEN, iHostPort, _stDirCfg::iRemoveNoise, _stDirReq2::iRemoveNoise, _stDirCfg::iScaling, _stDirReq2::iScaling, _stDirCfg::lGapLength, _stDirReq2::lGapLength, _stDirCfg::lMeanLength, _stDirReq2::lMeanLength, LOCATE_DIR_PROC, _stDirCfg::lWindowSize, _stDirReq2::lWindowSize, Msg, sDirCfg, clSockOp::SetHandle(), clDirMsg2::SetRequest(), SOp, and clSockOp::WriteN().

Referenced by Main().

7.102.3.4 int clSubLocate::Main (int, char **)

Definition at line 433 of file Locate.cc.

References bDebug, ConnectDir(), DirMsg, _stDirRes2::fFreqResolution, fpDirRes, fpLevRes, clDirMsg2::GetResult(), clLocateSensor::GetResultMatrix(), iDirMsgSize, Initialize(), lHeight, _stDirRes2::lMaxBin, _stDirRes2::lMinBin, LocSens, _stDirRes2::lResultCount, lWidth, MPICCtrl, MPICNormal, Msg, clMPIComm::ProbeNB(), clSockOp::ReadN(), clMPIComm::Recv(), RecvParams(), clMPIComm::Send(), clLocateSensor::SetDirectionValues(), clSockOp::SetRecvBufSize(), and SOp.

Referenced by main().

7.102.4 Member Data Documentation

7.102.4.1 int clSubLocate::iHostPort [private]

Definition at line 112 of file Locate.hh.

Referenced by ConnectDir(), and RecvParams().

7.102.4.2 int clSubLocate::iDirMsgSize [private]

Definition at line 113 of file Locate.hh.

Referenced by Initialize(), and Main().

7.102.4.3 long clSubLocate::lWidth [private]

Definition at line 114 of file Locate.hh.

Referenced by Initialize(), Main(), and RecvParams().

7.102.4.4 long clSubLocate::lHeight [private]

Definition at line 115 of file Locate.hh.

Referenced by Initialize(), Main(), and RecvParams().

7.102.4.5 long clSubLocate::lPosX [private]

Definition at line 116 of file Locate.hh.

Referenced by Initialize(), and RecvParams().

7.102.4.6 long clSubLocate::lPosY [private]

Definition at line 117 of file Locate.hh.

Referenced by Initialize(), and RecvParams().

7.102.4.7 float clSubLocate::fAzimuth [private]

Definition at line 118 of file Locate.hh.

Referenced by Initialize(), and RecvParams().

7.102.4.8 char clSubLocate::cpHostName[LOCATE_HOSTNAME_MAXLEN] [private]

Definition at line 119 of file Locate.hh.

Referenced by ConnectDir(), and RecvParams().

7.102.4.9 GDT* clSubLocate::fpLevRes [private]

Definition at line 120 of file Locate.hh.

Referenced by Initialize(), and Main().

7.102.4.10 GDT* clSubLocate::fpDirRes [private]

Definition at line 121 of file Locate.hh.

Referenced by Initialize(), and Main().

7.102.4.11 stDirCfg clSubLocate::sDirCfg [private]

Definition at line 122 of file Locate.hh.

Referenced by ConnectDir(), Initialize(), and RecvParams().

7.102.4.12 clAlloc clSubLocate::DirMsg [private]

Definition at line 123 of file Locate.hh.

Referenced by Initialize(), and Main().

7.102.4.13 clAlloc clSubLocate::LevRes [private]

Definition at line 124 of file Locate.hh.

Referenced by Initialize().

7.102.4.14 clAlloc clSubLocate::DirRes [private]

Definition at line 125 of file Locate.hh.

Referenced by Initialize().

7.102.4.15 clLocateSensor clSubLocate::LocSens [private]

Definition at line 126 of file Locate.hh.

Referenced by Initialize(), and Main().

7.102.4.16 clMPIComm clSubLocate::MPICCtrl [private]

Definition at line 127 of file Locate.hh.

Referenced by clSubLocate(), and Main().

7.102.4.17 clMPIComm clSubLocate::MPICParam [private]

Definition at line 128 of file Locate.hh.

Referenced by clSubLocate(), and RecvParams().

7.102.4.18 clMPIComm clSubLocate::MPICNormal [private]

Definition at line 129 of file Locate.hh.

Referenced by clSubLocate(), and Main().

7.102.4.19 clDirMsg2 clSubLocate::Msg [private]

Definition at line 130 of file Locate.hh.

Referenced by ConnectDir(), and Main().

7.102.4.20 clSockOp clSubLocate::SOp [private]

Definition at line 131 of file Locate.hh.

Referenced by ConnectDir(), and Main().

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

- **Locate.hh**
- **Locate.cc**

7.103 _stTBearInfo Struct Reference

Information stored in saved bearing-time .tif.inf.

`#include <ConvPic.hh>`

Public Attributes

- time_t **ttTime**

  Oldest dataline time.

- double **dLeftDir**

  Left angle (rad).

- double **dRightDir**

  Right angle (rad).

- double **dIntTime**

  Integration time (s).

- double **dHighFreq**

  Upper frequency limit.

- int **iSectors**

  Number of sectors.

7.103.1 Detailed Description

Information stored in saved bearing-time .tif.inf.

Definition at line 96 of file ConvPic.hh.

7.103.2 Member Data Documentation

7.103.2.1 time_t _stTBearInfo::ttTime

Oldest dataline time.

Definition at line 98 of file ConvPic.hh.

Referenced by clConvPic::DrawTBearInfo(), and clConvPic::LoadTBearInfo().

7.103.2.2 double _stTBearInfo::dLeftDir

Left angle (rad).

Definition at line 99 of file ConvPic.hh.

Referenced by clConvPic::DrawTBearInfo(), and clConvPic::LoadTBearInfo().

7.103.2.3 double _stTBearInfo::dRightDir

Right angle (rad).

Definition at line 100 of file ConvPic.hh.

Referenced by clConvPic::DrawTBearInfo(), and clConvPic::LoadTBearInfo().

7.103.2.4 double _stTBearInfo::dIntTime

Integration time (s).

Definition at line 101 of file ConvPic.hh.

Referenced by clConvPic::DrawTBearInfo(), and clConvPic::LoadTBearInfo().

7.103.2.5 double _stTBearInfo::dHighFreq

Upper frequency limit.

Definition at line 102 of file ConvPic.hh.

Referenced by clConvPic::LoadTBearInfo().

7.103.2.6 int _stTBearInfo::iSectors

Number of sectors.

Definition at line 103 of file ConvPic.hh.

Referenced by clConvPic::DrawTBearInfo(), and clConvPic::LoadTBearInfo().

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

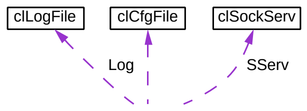
- **ConvPic.hh**

7.104 clUIServer Class Reference

User interface server.

`#include <UIServ.hh>`

Collaboration diagram for clUIServer:



Public Member Functions

- **clUIServer** ()
- ~**clUIServer** ()
- int **Wait** ()
- void **NoLogShutdown** ()

Public Attributes

- **clLogFile** ∗ **Log**

Private Attributes

- bool **bLogShutdown**
- **clCfgFile** ∗ **Cfg**
- **clSockServ** ∗ **SServ**

7.104.1 Detailed Description

User interface server.

This works like inetd. It starts waiting for connections in single predifined port. When connection arrives it receives process request from client, duplicates handles and fork() and exec() the requested process and then continues to wait for new connections.

There are several advatages of this behaviour:

- Takes only one port
- SMP friendly
- Easy to extend

Definition at line 47 of file UIServ.hh.

7.104.2 Constructor & Destructor Documentation

7.104.2.1 clUIServer::clUIServer ()

Definition at line 194 of file UIServ.cc.

References clLogFile::Add(), bLogShutdown, Cfg, clSockServ::GetErrno(), clLogFile::GetError(), clCfgFile::GetInt(), Log, LOGFILE_NOERROR, SServ, UIS_CFGFILE, UIS_CONV_BUF_SIZE, UIS_DEFAULT_PORT, and UIS_LOGFILE.

7.104.2.2 clUIServer::∼clUIServer ()

Definition at line 223 of file UIServ.cc.

References clLogFile::Add(), bLogShutdown, Cfg, Log, and SServ.

7.104.3 Member Function Documentation

7.104.3.1 int clUIServer::Wait ()

Definition at line 232 of file UIServ.cc.

References clLogFile::Add(), bRun, clSockServ::GetErrno(), clSockOp::GetPeerName(), Log, clSockOp::SetCloseOnDestruct(), clSockOp::SetHandle(), SOp, SServ, UIS_CONV_BUF_SIZE, UIS_SHUTDOWNFILE, UIS_TIMEOUT, and clSockServ::WaitForConnect().

Referenced by main().

7.104.3.2 void clUIServer::NoLogShutdown () `[inline]`

Definition at line 57 of file UIServ.hh.

References bLogShutdown.

Referenced by main().

7.104.4 Member Data Documentation

7.104.4.1 bool clUIServer::bLogShutdown `[private]`

Definition at line 49 of file UIServ.hh.

Referenced by clUIServer(), NoLogShutdown(), and ∼clUIServer().

7.104.4.2 clCfgFile∗ clUIServer::Cfg `[private]`

Definition at line 50 of file UIServ.hh.

Referenced by clUIServer(), and ~clUIServer().

7.104.4.3 clSockServ* clUIServer::SServ [private]

Definition at line 51 of file UIServ.hh.

Referenced by clUIServer(), Wait(), and ~clUIServer().

7.104.4.4 clLogFile* clUIServer::Log

Definition at line 53 of file UIServ.hh.

Referenced by clUIServer(), Wait(), and ~clUIServer().

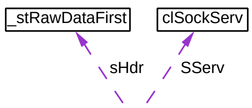
The documentation for this class was generated from the following files:

- **UIServ.hh**
- **UIServ.cc**

7.105 clXMMSOut Class Reference

`#include <XMMSOut.hh>`

Collaboration diagram for clXMMSOut:



Public Member Functions

- **clXMMSOut** ()
- ~**clXMMSOut** ()
- void **Init** ()
- void **About** ()
- void **Configure** ()
- int **OpenAudio** (AFormat, int, int)
- void **WriteAudio** (void *, int)
- void **CloseAudio** ()
- void **Flush** (int)
- void **Pause** (short)
- int **BufferFree** ()
- int **BufferPlaying** ()
- int **OutputTime** ()
- int **WrittenTime** ()
- void * **MainThread** (void *)
- void * **ServeClientThread** (void *)
- void **Stop** ()
- void **OnAboutButton** (GtkButton *, gpointer)
- void **OnButtonClick** (GtkButton *, gpointer)

Private Member Functions

- double **GetTime** ()
- void **Convert8s8u** (void *, int)
- void **Convert16u16s** (void *, int)
- void **EndianConvert** (unsigned short *, int)
- void **CopyChannel** (GDT *, const GDT *, int)

Private Attributes

- volatile bool **bRun**
- volatile bool **bPause**
- int **iMainThreadH**
- int **iWriteTime**
- int **iPlayTime**
- volatile int **iAudioBufSize**
- volatile int **iFragmentSize**
- unsigned long long **uiTotalTickCount**
- double **dStartTime**
- char * **cpLocalSocket**
- AFormat **eAudioFormat**
- **stRawDataFirst sHdr**
- clReBufferT< GDT > **ReBuffer**
- clAlloc **AudioBuf**
- clMutex **MtxAudio**
- clCondition **CndAudio**
- clDSPOp **DSP**
- **clSockServ SServ**
- gchar * **cpMessageTxt**
- GtkWidget * **gwMessageBox**
- GtkWidget * **gwWinConfig**
- GtkWidget * **gwVBox**
- GtkWidget * **gwHBox**
- GtkWidget * **gwLBufSize**
- GtkWidget * **gwEBufSize**
- GtkWidget * **gwLLocalSocket**
- GtkWidget * **gwELocalSocket**
- GtkWidget * **gwBOk**
- GtkWidget * **gwBCancel**

7.105.1 Constructor & Destructor Documentation

7.105.1.1 clXMMSOut::clXMMSOut ()

Definition at line 253 of file XMMSOut.cc.

References bPause, bRun, cpLocalSocket, dStartTime, gwWinConfig, iAudioBufSize, iFragmentSize, and iMainThreadH.

7.105.1.2 clXMMSOut::~clXMMSOut ()

Definition at line 266 of file XMMSOut.cc.

References CloseAudio().

7.105.2 Member Function Documentation

7.105.2.1 double clXMMSOut::GetTime () [inline, private]

Definition at line 186 of file XMMSOut.cc.

Referenced by Flush(), and WriteAudio().

7.105.2.2 void clXMMSOut::Convert8s8u (void *, int) [inline, private]

Definition at line 197 of file XMMSOut.cc.

Referenced by WriteAudio().

7.105.2.3 void clXMMSOut::Convert16u16s (void *, int) [inline, private]

Definition at line 211 of file XMMSOut.cc.

Referenced by WriteAudio().

7.105.2.4 void clXMMSOut::EndianConvert (unsigned short *, int) [inline, private]

Definition at line 225 of file XMMSOut.cc.

7.105.2.5 void clXMMSOut::CopyChannel (GDT *, const GDT *, int) [inline, private]

Definition at line 239 of file XMMSOut.cc.

References _stRawDataFirst::iChannels, iFragmentSize, and sHdr.

Referenced by ServeClientThread().

7.105.2.6 void clXMMSOut::Init ()

Definition at line 272 of file XMMSOut.cc.

References AudioBuf, clSockServ::Bind(), cpDefLocalSocket, cpLocalSocket, GDT, iAudioBufSize, iDefBufferSize, iFragmentSize, and SServ.

Referenced by hasas_init(), and OnButtonClick().

7.105.2.7 void clXMMSOut::About ()

Definition at line 298 of file XMMSOut.cc.

References cpMessageTxt, GLOBAL_VERSMAJ, GLOBAL_VERSMIN, GLOBAL_VERSPL, gwMessageBox, WrapOnAboutButton(), and XMMSOUT_DESCRIPTION.

Referenced by hasas_about().

7.105.2.8 void clXMMSOut::Configure ()

Definition at line 309 of file XMMSOut.cc.

References cpBCancelTxt, cpBOkTxt, cpLBufSizeTxt, cpLLocalSocketTxt, cpLocalSocket, cpWinTitleTxt, gwBCancel, gwBOk, gwEBufSize, gwELocalSocket, gwHBox, gwLBufSize, gwLLocalSocket, gwVBox, gwWinConfig, iAudioBufSize, and WrapOnButtonClick().

Referenced by hasas_configure().

7.105.2.9 int clXMMSOut::OpenAudio (AFormat, int, int)

Definition at line 370 of file XMMSOut.cc.

References bRun, _stRawDataFirst::dSampleRate, dStartTime, eAudioFormat, _stRawDataFirst::iChannels, iMainThreadH, iPlayTime, iWriteTime, MainThread(), sHdr, uiTotalTickCount, and XMMSOutThreads().

Referenced by hasas_open_audio().

7.105.2.10 void clXMMSOut::WriteAudio (void *, int)

Definition at line 392 of file XMMSOut.cc.

References AudioBuf, bPause, CndAudio, Convert16u16s(), Convert8s8u(), _stRawDataFirst::dSampleRate, DSP, dStartTime, eAudioFormat, GDT, GetTime(), _stRawDataFirst::iChannels, iFragmentSize, iPlayTime, iWriteTime, MtxAudio, ReBuffer, sHdr, and uiTotalTickCount.

Referenced by hasas_write_audio().

7.105.2.11 void clXMMSOut::CloseAudio ()

Definition at line 489 of file XMMSOut.cc.

References iMainThreadH, Stop(), and XMMSOutThreads().

Referenced by hasas_close_audio(), and ~clXMMSOut().

7.105.2.12 void clXMMSOut::Flush (int)

Definition at line 500 of file XMMSOut.cc.

References dStartTime, GetTime(), iPlayTime, iWriteTime, and uiTotalTickCount.

Referenced by hasas_flush().

7.105.2.13 void clXMMSOut::Pause (short)

Definition at line 509 of file XMMSOut.cc.

References bPause.

Referenced by hasas_pause().

7.105.2.14 int clXMMSOut::BufferFree ()

Definition at line 515 of file XMMSOut.cc.

References iAudioBufSize.

Referenced by hasas_buffer_free().

7.105.2.15 int clXMMSOut::BufferPlaying ()

Definition at line 522 of file XMMSOut.cc.

Referenced by hasas_buffer_playing().

7.105.2.16 int clXMMSOut::OutputTime ()

Definition at line 529 of file XMMSOut.cc.

References iPlayTime.

Referenced by hasas_output_time().

7.105.2.17 int clXMMSOut::WrittenTime ()

Definition at line 535 of file XMMSOut.cc.

References iWriteTime.

Referenced by hasas_written_time().

7.105.2.18 void * clXMMSOut::MainThread (void *)

Definition at line 541 of file XMMSOut.cc.

References bRun, ServeClientThread(), SServ, clSockServ::WaitForConnect(), and XMMSOutThreads().

Referenced by OpenAudio().

7.105.2.19 void * clXMMSOut::ServeClientThread (void *)

Definition at line 560 of file XMMSOut.cc.

References AudioBuf, bRun, CndAudio, CopyChannel(), DSP, GDT, iAudioBufSize, _stRawDataReq::iChannel, _stRawDataFirst::iChannels, iFragmentSize, MtxAudio, clSockOp::ReadN(), sHdr, SOp, and clSockOp::WriteN().

Referenced by MainThread().

7.105.2.20 void clXMMSOut::Stop () [inline]

Definition at line 103 of file XMMSOut.hh.

References bRun.

Referenced by CloseAudio().

7.105.2.21 void clXMMSOut::OnAboutButton (GtkButton *, gpointer)

Definition at line 605 of file XMMSOut.cc.

References cpMessageTxt, and gwMessageBox.

Referenced by WrapOnAboutButton().

7.105.2.22 void clXMMSOut::OnButtonClick (GtkButton *, gpointer)

Definition at line 612 of file XMMSOut.cc.

References clSockServ::Close(), gwBCancel, gwBOk, gwEBufSize, gwELocalSocket, gwWinConfig, Init(), and SServ.

Referenced by WrapOnButtonClick().

7.105.3 Member Data Documentation

7.105.3.1 volatile bool clXMMSOut::bRun [private]

Definition at line 50 of file XMMSOut.hh.

Referenced by clXMMSOut(), MainThread(), OpenAudio(), ServeClientThread(), and Stop().

7.105.3.2 volatile bool clXMMSOut::bPause [private]

Definition at line 51 of file XMMSOut.hh.

Referenced by clXMMSOut(), Pause(), and WriteAudio().

7.105.3.3 int clXMMSOut::iMainThreadH [private]

Definition at line 52 of file XMMSOut.hh.

Referenced by CloseAudio(), clXMMSOut(), and OpenAudio().

7.105.3.4 int clXMMSOut::iWriteTime [private]

Definition at line 53 of file XMMSOut.hh.

Referenced by Flush(), OpenAudio(), WriteAudio(), and WrittenTime().

7.105.3.5 int clXMMSOut::iPlayTime [private]

Definition at line 54 of file XMMSOut.hh.

Referenced by Flush(), OpenAudio(), OutputTime(), and WriteAudio().

7.105.3.6 volatile int clXMMSOut::iAudioBufSize [private]

Definition at line 55 of file XMMSOut.hh.

Referenced by BufferFree(), clXMMSOut(), Configure(), Init(), and ServeClientThread().

7.105.3.7 volatile int clXMMSOut::iFragmentSize [private]

Definition at line 56 of file XMMSOut.hh.

Referenced by clXMMSOut(), CopyChannel(), Init(), ServeClientThread(), and WriteAudio().

7.105.3.8 unsigned long long clXMMSOut::uiTotalTickCount [private]

Definition at line 57 of file XMMSOut.hh.

Referenced by Flush(), OpenAudio(), and WriteAudio().

7.105.3.9 double clXMMSOut::dStartTime [private]

Definition at line 58 of file XMMSOut.hh.

Referenced by clXMMSOut(), Flush(), OpenAudio(), and WriteAudio().

7.105.3.10 char* clXMMSOut::cpLocalSocket [private]

Definition at line 59 of file XMMSOut.hh.

Referenced by clXMMSOut(), Configure(), and Init().

7.105.3.11 AFormat clXMMSOut::eAudioFormat [private]

Definition at line 60 of file XMMSOut.hh.

Referenced by OpenAudio(), and WriteAudio().

7.105.3.12 stRawDataFirst clXMMSOut::sHdr [private]

Definition at line 61 of file XMMSOut.hh.

Referenced by CopyChannel(), OpenAudio(), ServeClientThread(), and WriteAudio().

7.105.3.13 clReBufferT<GDT> clXMMSOut::ReBuffer [private]

Definition at line 62 of file XMMSOut.hh.

Referenced by WriteAudio().

7.105.3.14 clAlloc clXMMSOut::AudioBuf [private]

Definition at line 63 of file XMMSOut.hh.

Referenced by Init(), ServeClientThread(), and WriteAudio().

7.105.3.15 clMutex clXMMSOut::MtxAudio [private]

Definition at line 64 of file XMMSOut.hh.

Referenced by ServeClientThread(), and WriteAudio().

7.105.3.16 clCondition clXMMSOut::CndAudio [private]

Definition at line 65 of file XMMSOut.hh.

Referenced by ServeClientThread(), and WriteAudio().

7.105.3.17 clDSPOp clXMMSOut::DSP [private]

Definition at line 66 of file XMMSOut.hh.

Referenced by ServeClientThread(), and WriteAudio().

7.105.3.18 clSockServ clXMMSOut::SServ [private]

Definition at line 67 of file XMMSOut.hh.

Referenced by Init(), MainThread(), and OnButtonClick().

7.105.3.19 gchar∗ clXMMSOut::cpMessageTxt [private]

Definition at line 69 of file XMMSOut.hh.

Referenced by About(), and OnAboutButton().

7.105.3.20 GtkWidget∗ clXMMSOut::gwMessageBox [private]

Definition at line 70 of file XMMSOut.hh.

Referenced by About(), and OnAboutButton().

7.105.3.21 GtkWidget∗ clXMMSOut::gwWinConfig [private]

Definition at line 71 of file XMMSOut.hh.

Referenced by clXMMSOut(), Configure(), and OnButtonClick().

7.105.3.22 GtkWidget∗ clXMMSOut::gwVBox [private]

Definition at line 72 of file XMMSOut.hh.

Referenced by Configure().

7.105.3.23 GtkWidget∗ clXMMSOut::gwHBox [private]

Definition at line 73 of file XMMSOut.hh.

Referenced by Configure().

7.105.3.24 GtkWidget∗ clXMMSOut::gwLBufSize [private]

Definition at line 74 of file XMMSOut.hh.

Referenced by Configure().

7.105.3.25 GtkWidget∗ clXMMSOut::gwEBufSize [private]

Definition at line 75 of file XMMSOut.hh.

Referenced by Configure(), and OnButtonClick().

7.105.3.26 GtkWidget∗ clXMMSOut::gwLLocalSocket [private]

Definition at line 76 of file XMMSOut.hh.

Referenced by Configure().

7.105.3.27 GtkWidget* clXMMSOut::gwELocalSocket [private]

Definition at line 77 of file XMMSOut.hh.

Referenced by Configure(), and OnButtonClick().

7.105.3.28 GtkWidget* clXMMSOut::gwBOk [private]

Definition at line 78 of file XMMSOut.hh.

Referenced by Configure(), and OnButtonClick().

7.105.3.29 GtkWidget* clXMMSOut::gwBCancel [private]

Definition at line 79 of file XMMSOut.hh.

Referenced by Configure(), and OnButtonClick().

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

- **XMMSOut.hh**
- **XMMSOut.cc**

Chapter 8

HASAS File Documentation

8.1 ArrayBase.cc File Reference

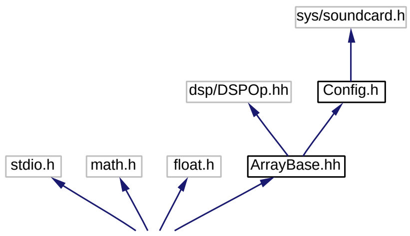
`#include <stdio.h>`

`#include <math.h>`

`#include <float.h>`

`#include "ArrayBase.hh"`

Include dependency graph for ArrayBase.cc:



Variables

- const char ∗ **cpaShadingTypes** []

8.1.1 Variable Documentation

8.1.1.1 const char* cpaShadingTypes[] [static]

Initial value:

```
{ "Rectangle", "Blackman",
   "Kaiser-Bessel" }
```

Definition at line 30 of file ArrayBase.cc.

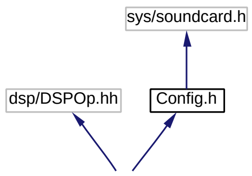
Referenced by clArrayBase::SetShading().

8.2 ArrayBase.hh File Reference

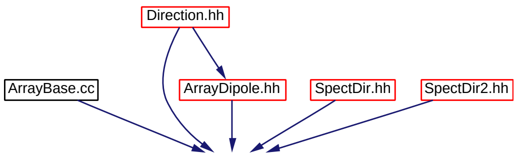
`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

Include dependency graph for ArrayBase.hh:



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



Compounds

- class **clArrayBase**

  Base class for array operations.

Enumerations

- enum { **AB_SHADE_RECTANGLE** = 0, **AB_SHADE_BLACKMAN** = 1, **AB_SHADE_KAISER_BESSEL** = 2 }

  Types for shading window.

8.2.1 Enumeration Type Documentation

8.2.1.1 anonymous enum

Types for shading window.

Enumeration values:

AB_SHADE_RECTANGLE

AB_SHADE_BLACKMAN
AB_SHADE_KAISER_BESSEL

Definition at line 35 of file ArrayBase.hh.

8.3 ArrayDipole.cc File Reference

`#include <stdio.h>`

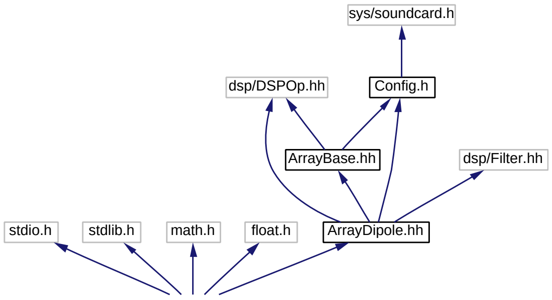
`#include <stdlib.h>`

`#include <math.h>`

`#include <float.h>`

`#include "ArrayDipole.hh"`

Include dependency graph for ArrayDipole.cc:



8.4 ArrayDipole.hh File Reference

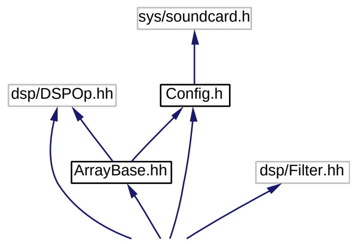
`#include <dsp/DSPOp.hh>`

`#include <dsp/Filter.hh>`

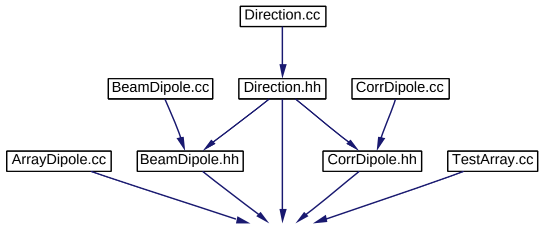
`#include "Config.h"`

`#include "ArrayBase.hh"`

Include dependency graph for ArrayDipole.hh:



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



Compounds

- class **clArrayDipole**
  Class for dipole array operations.

8.5 ArraySensor.cc File Reference

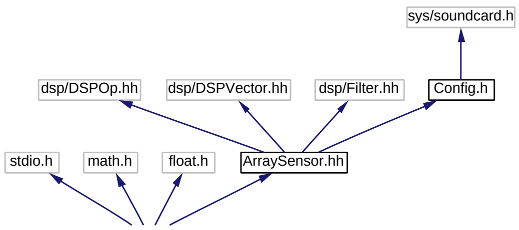
#include <stdio.h>

#include <math.h>

#include <float.h>

#include "ArraySensor.hh"

Include dependency graph for ArraySensor.cc:



8.6 ArraySensor.hh File Reference

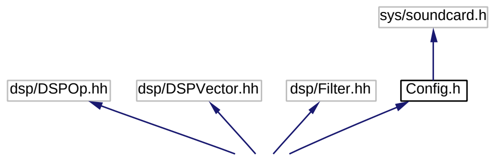
`#include <dsp/DSPOp.hh>`

`#include <dsp/DSPVector.hh>`

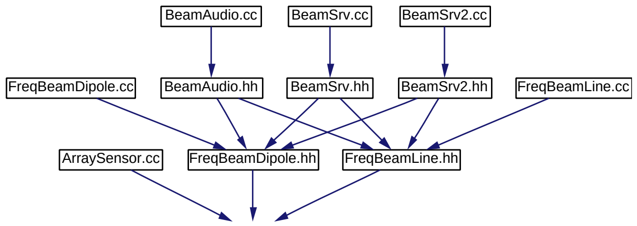
`#include <dsp/Filter.hh>`

`#include "Config.h"`

Include dependency graph for ArraySensor.hh:



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



Compounds

- class **clArraySensor**

  Class representing sensor in sensor array for frequency domain beamforming.

8.7 Audio.cc File Reference

`#include <stdio.h>`

`#include <stdlib.h>`

`#include <unistd.h>`

`#include <fcntl.h>`

`#include <ioctl.h>`

`#include <errno.h>`

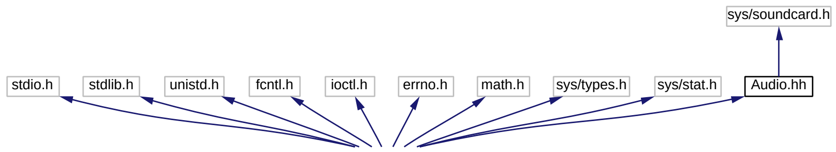
`#include <math.h>`

`#include <sys/types.h>`

`#include <sys/stat.h>`

`#include "Audio.hh"`

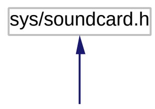
Include dependency graph for Audio.cc:



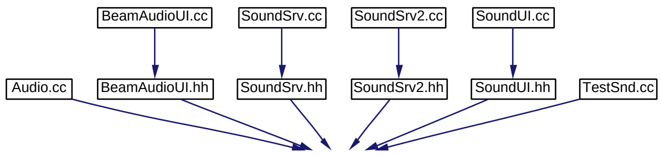
8.8 Audio.hh File Reference

`#include "sys/soundcard.h"`

Include dependency graph for Audio.hh:



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



Compounds

- class **clAudio**

  Class for OSS audio IO operations.

Defines

- #define **BUF_SIZE** 4096

Enumerations

- enum { **AUDIO_READ** = 0, **AUDIO_WRITE** = 1, **AUDIO_DUPLEX** = 2 }

8.8.1 Define Documentation

8.8.1.1 #define BUF_SIZE 4096

Definition at line 34 of file Audio.hh.

8.8.2 Enumeration Type Documentation

8.8.2.1 anonymous enum

Enumeration values:

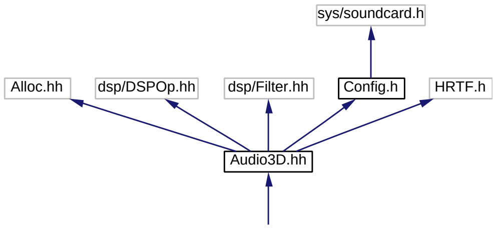
- **AUDIO_READ**
- **AUDIO_WRITE**
- **AUDIO_DUPLEX**

Definition at line 38 of file Audio.hh.

8.9 Audio3D.cc File Reference

`#include "Audio3D.hh"`

Include dependency graph for Audio3D.cc:



Defines

- #define **COMPILE**

8.9.1 Define Documentation

8.9.1.1 #define COMPILE

Definition at line 23 of file Audio3D.cc.

8.10 Audio3D.hh File Reference

`#include <Alloc.hh>`

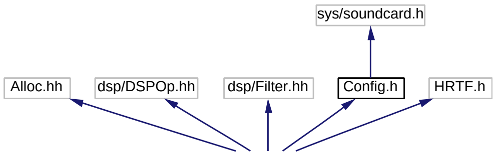
`#include <dsp/DSPOp.hh>`

`#include <dsp/Filter.hh>`

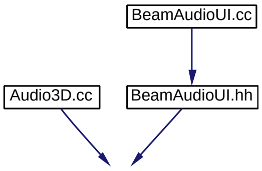
`#include "Config.h"`

`#include "HRTF.h"`

Include dependency graph for Audio3D.hh:



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



Compounds

- class **clAudio3D**

  3D audio engine

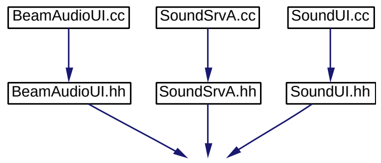
8.11 AudioA.cc File Reference

8.12 AudioA.hh File Reference

8.13 AudioA2.cc File Reference

8.14 AudioA2.hh File Reference

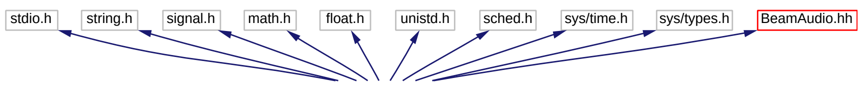
This graph shows which files directly or indirectly include this file:



8.15 BeamAudio.cc File Reference

```
#include <stdio.h>
#include <string.h>
#include <signal.h>
#include <math.h>
#include <float.h>
#include <unistd.h>
#include <sched.h>
#include <sys/time.h>
#include <sys/types.h>
#include "BeamAudio.hh"
```

Include dependency graph for BeamAudio.cc:



Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char ∗argv[ ])

Variables

- bool **bDebug**
- bool **bDaemon**
- **clBeamAudio** ∗ **BeamAudio**

8.15.1 Function Documentation

8.15.1.1 void SigHandler (int *iSigNum*)

Definition at line 41 of file BeamAudio.cc.

References bDebug.

Referenced by clStreamDist::Main(), clSoundSrvA::Main(), clSoundSrv2::Main(), clComediSrv::Main(), and main().

8.15.1.2 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 57 of file BeamAudio.cc.

References BA_SCHED_PRIORITY, bDaemon, bDebug, BeamAudio, clBeamAudio::Main(), and SigHandler().

8.15.2 Variable Documentation

8.15.2.1 bool bDebug `[static]`

Definition at line 36 of file BeamAudio.cc.

Referenced by clBeamAudio::InitBeam(), clBeamAudio::Main(), main(), and SigHandler().

8.15.2.2 bool bDaemon `[static]`

Definition at line 37 of file BeamAudio.cc.

Referenced by main().

8.15.2.3 clBeamAudio∗ BeamAudio

Definition at line 38 of file BeamAudio.cc.

Referenced by main().

8.16 BeamAudio.hh File Reference

`#include <limits.h>`

`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

`#include "LocalMsg.h"`

`#include "CfgFile.hh"`

`#include "FreqBeamDipole.hh"`

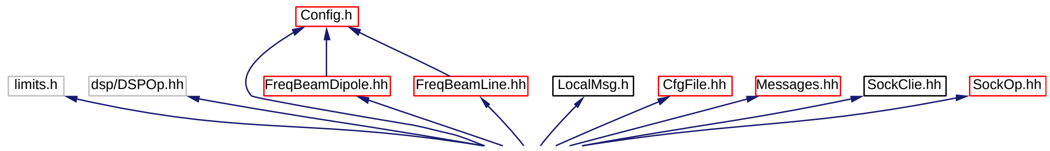
`#include "FreqBeamLine.hh"`

`#include "Messages.hh"`

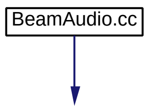
`#include "SockClie.hh"`

`#include "SockOp.hh"`

Include dependency graph for BeamAudio.hh:



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



Compounds

- class **clBeamAudio**

  Server for beamformed audio.

Enumerations

- enum {

  BA_ARRAY_TYPE_DIPOLE = 0, **BA_ARRAY_TYPE_TRIANGLE** = 1, **BA_ARRAY_TYPE_LINE** = 2, **BA_ARRAY_TYPE_PLANE** = 3,

BA_ARRAY_TYPE_CYLINDER = 4, **BA_ARRAY_TYPE_-SPHERE** = 5 }

8.16.1 Enumeration Type Documentation

8.16.1.1 anonymous enum

Enumeration values:

BA_ARRAY_TYPE_DIPOLE

BA_ARRAY_TYPE_TRIANGLE

BA_ARRAY_TYPE_LINE

BA_ARRAY_TYPE_PLANE

BA_ARRAY_TYPE_CYLINDER

BA_ARRAY_TYPE_SPHERE

Definition at line 41 of file BeamAudio.hh.

8.17 BeamAudioUI.cc File Reference

`#include <pthread.h>`

`#include <stdio.h>`

`#include <stdlib.h>`

`#include <math.h>`

`#include <float.h>`

`#include <string.h>`

`#include <limits.h>`

`#include <unistd.h>`

`#include <signal.h>`

`#include <fcntl.h>`

`#include <sched.h>`

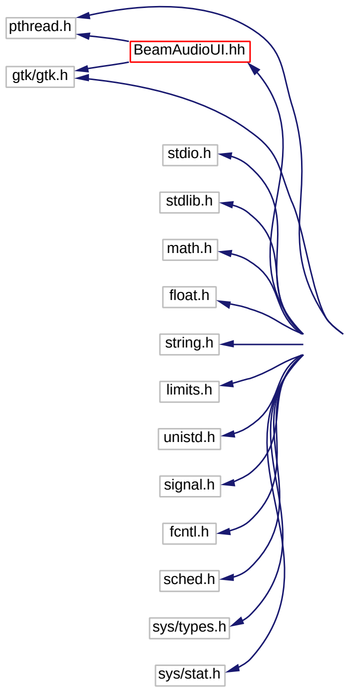
`#include <sys/types.h>`

`#include <sys/stat.h>`

`#include <gtk/gtk.h>`

`#include "BeamAudioUI.hh"`

Include dependency graph for BeamAudioUI.cc:



Functions

- **G_LOCK_DEFINE_STATIC** (gmInputMutex)
- int **main** (int argc, char ∗argv[])
- gboolean **WrapOnDelete** (GtkWidget ∗gwSender, GdkEvent ∗geEvent, gpointer gpData)
- void **WrapOnConnectClick** (GtkButton ∗gbButton, gpointer gpData)
- void **WrapOnValueChanged** (GtkAdjustment ∗gaAdjustment, gpointer gpData)
- void **WrapOnToggled** (GtkToggleButton ∗gtbToggleButton, gpointer gpData)
- void **WrapOnGdkInput** (gpointer gpData, gint giSource, GdkInputCondition gicCondition)
- void ∗ **WrapAudioOutThread** (void ∗vpParam)

Variables

- **clBeamAudioUI** ∗ **BeamAudioUI**

8.17.1 Function Documentation

8.17.1.1 G_LOCK_DEFINE_STATIC (gmInputMutex)

8.17.1.2 int main (int *argc*, char ∗ *argv*[])

Definition at line 46 of file BeamAudioUI.cc.

References BeamAudioUI, and clBeamAudioUI::Main().

8.17.1.3 gboolean WrapOnDelete (GtkWidget ∗ *gwSender*, GdkEvent ∗ *geEvent*, gpointer *gpData*)

Definition at line 59 of file BeamAudioUI.cc.

References BeamAudioUI, and clBeamAudioUI::OnDelete().

Referenced by clGUITransient::ConnectSignals(), clSpectGUI::ConnectSignals(), clGUILofar::ConnectSignals(), clGUILocate::ConnectSignals(), clGUILevel::ConnectSignals(), clGUIDir::ConnectSignals(), clFileSrv::ConnectSignals(), and clBeamAudioUI::ConnectSignals().

8.17.1.4 void WrapOnConnectClick (GtkButton ∗ *gbButton*, gpointer *gpData*)

Definition at line 65 of file BeamAudioUI.cc.

References BeamAudioUI, and clBeamAudioUI::OnConnectClick().

Referenced by clGUITransient::ConnectSignals(), clSpectGUI::ConnectSignals(), clGUILofar::ConnectSignals(), clGUILocate::ConnectSignals(), clGUILevel::ConnectSignals(), clGUIDir::ConnectSignals(), and clBeamAudioUI::ConnectSignals().

8.17.1.5 void WrapOnValueChanged (GtkAdjustment * *gaAdjustment*, gpointer *gpData*)

Definition at line 71 of file BeamAudioUI.cc.

References BeamAudioUI, and clBeamAudioUI::OnValueChanged().

Referenced by clBeamAudioUI::ConnectSignals().

8.17.1.6 void WrapOnToggled (GtkToggleButton * *gtbToggleButton*, gpointer *gpData*)

Definition at line 77 of file BeamAudioUI.cc.

References BeamAudioUI, and clBeamAudioUI::OnToggled().

Referenced by clBeamAudioUI::ConnectSignals().

8.17.1.7 void WrapOnGdkInput (gpointer *gpData*, gint *giSource*, GdkInputCondition *gicCondition*)

Definition at line 83 of file BeamAudioUI.cc.

References BeamAudioUI, and clBeamAudioUI::OnGdkInput().

Referenced by clGUILofar::InitConnection(), clGUIDir::InitConnection(), clBeamAudioUI::InitConnection(), clGUILocate::OnConnectClick(), and clGUILevel::OnConnectClick().

8.17.1.8 void* WrapAudioOutThread (void * *vpParam*)

Definition at line 90 of file BeamAudioUI.cc.

References clBeamAudioUI::AudioOutThread(), and BeamAudioUI.

Referenced by clBeamAudioUI::SendSettings().

8.17.2 Variable Documentation

8.17.2.1 clBeamAudioUI* BeamAudioUI

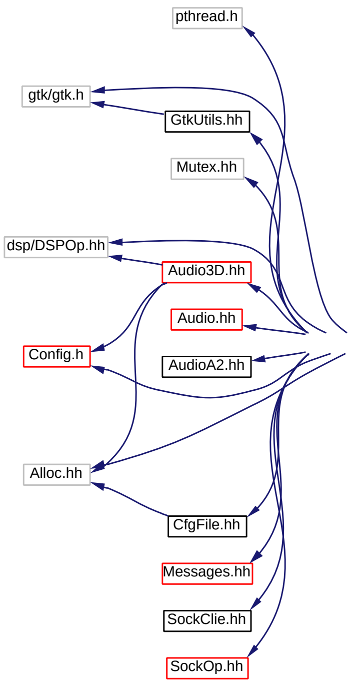
Definition at line 43 of file BeamAudioUI.cc.

Referenced by main(), WrapAudioOutThread(), WrapOnConnectClick(), WrapOnDelete(), WrapOnGdkInput(), WrapOnToggled(), and WrapOnValueChanged().

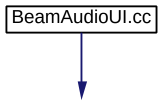
8.18 BeamAudioUI.hh File Reference

```
#include <pthread.h>
#include <gtk/gtk.h>
#include <Alloc.hh>
#include <Mutex.hh>
#include <dsp/DSPOp.hh>
#include "Config.h"
#include "Audio.hh"
#include "AudioA2.hh"
#include "Audio3D.hh"
#include "CfgFile.hh"
#include "GtkUtils.hh"
#include "Messages.hh"
#include "SockClie.hh"
#include "SockOp.hh"
```

Include dependency graph for BeamAudioUI.hh:



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



Compounds

- class **clBeamAudioUI**

  User interface for beam audio server.

Defines

- #define **BAUI_VER_MAJ** GLOBAL_VERSMAJ
- #define **BAUI_VER_MIN** GLOBAL_VERSMIN
- #define **BAUI_VER_PL** GLOBAL_VERSPL
- #define **BAUI_PADDING** 8
- #define **BAUI_SERVER_MAXLEN** 255
- #define **BAUI_CONV_BUF_LEN** 255
- #define **BAUI_AUDIO_BUFCOUNT** 10

Variables

- const char * **cpWindowTxt** = "Beam audio UI"
- const char * **cpLServerTxt** = "Server"
- const char * **cpBConnectTxt** = "Connect"
- const char * **cpLSoundSpeedTxt** = "Sound speed"
- const char * **cpCBHighFreqTxt** = "High frequencies"
- const char * **cpCBDitherTxt** = "Dither"
- const char * **cpCB3DAudioTxt** = "3D"
- const char * **cpLDirectionTxt** = "Direction"

8.18.1 Define Documentation

8.18.1.1 #define BAUI_VER_MAJ GLOBAL_VERSMAJ

Definition at line 53 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI::clBeamAudioUI().

8.18.1.2 #define BAUI_VER_MIN GLOBAL_VERSMIN

Definition at line 54 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI::clBeamAudioUI().

8.18.1.3 #define BAUI_VER_PL GLOBAL_VERSPL

Definition at line 55 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI::clBeamAudioUI().

8.18.1.4 #define BAUI_PADDING 8

Definition at line 56 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI::Build(), clBeamAudioUI::BuildTable1(), and clBeamAudioUI::BuildTable2().

8.18.1.5 #define BAUI_SERVER_MAXLEN 255

Definition at line 57 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI::BuildTable1(), clBeamAudioUI::OnConnectClick(), and clBeamAudioUI::ParseServerStr().

8.18.1.6 #define BAUI_CONV_BUF_LEN 255

Definition at line 58 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI::OnGdkInput().

8.18.1.7 #define BAUI_AUDIO_BUFCOUNT 10

Definition at line 59 of file BeamAudioUI.hh.

Referenced by clBeamAudioUI::AllocateBuffers(), clBeamAudioUI::AudioOutThread(), and clBeamAudioUI::OnGdkInput().

8.18.2 Variable Documentation

8.18.2.1 const char* cpWindowTxt = "Beam audio UI" [static]

Definition at line 62 of file BeamAudioUI.hh.

8.18.2.2 const char* cpLServerTxt = "Server" [static]

Definition at line 63 of file BeamAudioUI.hh.

8.18.2.3 const char* cpBConnectTxt = "Connect" [static]

Definition at line 64 of file BeamAudioUI.hh.

8.18.2.4 const char* cpLSoundSpeedTxt = "Sound speed" [static]

Definition at line 65 of file BeamAudioUI.hh.

8.18.2.5 const char* cpCBHighFreqTxt = "High frequencies" [static]

Definition at line 66 of file BeamAudioUI.hh.

8.18.2.6 const char* cpCBDitherTxt = "Dither" [static]

Definition at line 67 of file BeamAudioUI.hh.

8.18.2.7 const char* cpCB3DAudioTxt = "3D" [static]

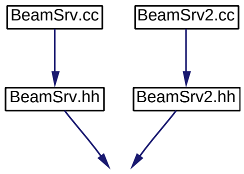
Definition at line 68 of file BeamAudioUI.hh.

8.18.2.8 const char* cpLDirectionTxt = "Direction" [static]

Definition at line 69 of file BeamAudioUI.hh.

8.19 BeamCommon.hh File Reference

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



Compounds

- struct **_stBeamNodeParams**

  Node parameters for beamforming cluster.

Defines

- #define **BS_LOGBUFSIZE** 4096
- #define **BS_ACCEPT_TO** 250

Typedefs

- typedef **_stBeamNodeParams stBeamNodeParams**

  Node parameters for beamforming cluster.

- typedef **_stBeamNodeParams** ∗ **stpBeamNodeParams**

  Node parameters for beamforming cluster.

Enumerations

- enum {

  BS_ARRAY_TYPE_DIPOLE = 0, **BS_ARRAY_TYPE_TRIANGLE** = 1, **BS_ARRAY_TYPE_LINE** = 2, **BS_ARRAY_TYPE_PLANE** = 3,

  BS_ARRAY_TYPE_CYLINDER = 4, **BS_ARRAY_TYPE_SPHERE** = 5 }

  Array types for beamforming server.

8.19.1 Define Documentation

8.19.1.1 #define BS_LOGBUFSIZE 4096

Definition at line 26 of file BeamCommon.hh.

8.19.1.2 #define BS_ACCEPT_TO 250

Definition at line 27 of file BeamCommon.hh.

Referenced by clBeamSrv2Master::ServerThread(), and clBeamSrvMaster::ServerThread().

8.19.2 Typedef Documentation

8.19.2.1 typedef struct _stBeamNodeParams stBeamNodeParams

Node parameters for beamforming cluster.

8.19.2.2 typedef struct _stBeamNodeParams * stpBeamNodeParams

Node parameters for beamforming cluster.

8.19.3 Enumeration Type Documentation

8.19.3.1 anonymous enum

Array types for beamforming server.

Enumeration values:

- **BS_ARRAY_TYPE_DIPOLE**
- **BS_ARRAY_TYPE_TRIANGLE**
- **BS_ARRAY_TYPE_LINE**
- **BS_ARRAY_TYPE_PLANE**
- **BS_ARRAY_TYPE_CYLINDER**
- **BS_ARRAY_TYPE_SPHERE**

Definition at line 33 of file BeamCommon.hh.

8.20 BeamDipole.cc File Reference

`#include <stdio.h>`

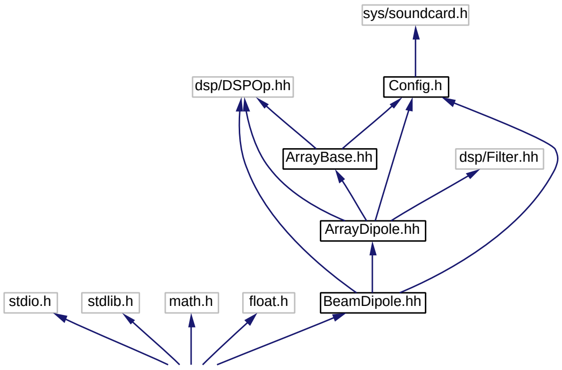
`#include <stdlib.h>`

`#include <math.h>`

`#include <float.h>`

`#include "BeamDipole.hh"`

Include dependency graph for BeamDipole.cc:



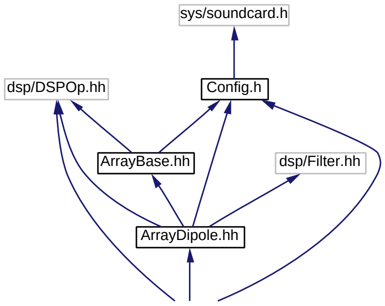
8.21 BeamDipole.hh File Reference

`#include <dsp/DSPOp.hh>`

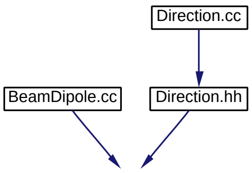
`#include "Config.h"`

`#include "ArrayDipole.hh"`

Include dependency graph for BeamDipole.hh:



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



Compounds

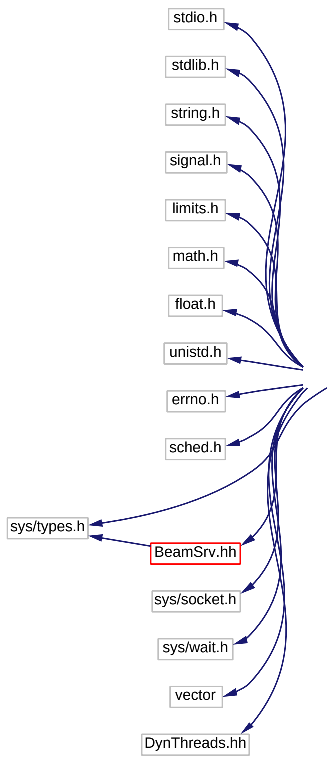
- class **clBeamDipole**

  Beamformer class for dipole array.

8.22 BeamSrv.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <limits.h>
#include <math.h>
#include <float.h>
#include <unistd.h>
#include <errno.h>
#include <sched.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/wait.h>
#include <vector>
#include <DynThreads.hh>
#include "BeamSrv.hh"
```

Include dependency graph for BeamSrv.cc:



Functions

- void **sig_handler_m** (int signo)
- void **sig_handler_s** (int signo)
- int **main** (int argc, char ∗argv[ ])

Variables

- volatile bool **bRun** = true
- clDynThreads< **clBeamSrvMaster** > ∗ **BeamSrvMasterThreads**
- BS_PROCINFOV_T **vProcInfo**

8.22.1 Function Documentation

8.22.1.1 void sig_handler_m (int *signo*)

Definition at line 49 of file BeamSrv.cc.

References bRun.

Referenced by main().

8.22.1.2 void sig_handler_s (int *signo*)

Definition at line 62 of file BeamSrv.cc.

Referenced by main().

8.22.1.3 int main (int *argc*, char * *argv*[ ])

Definition at line 75 of file BeamSrv.cc.

References BeamSrvMasterThreads, GLOBAL_VERSMAJ, GLOBAL_VERSMIN, GLOBAL_VERSPL, _stBeamProcInfo::iSockH, clBeamSrvSlave::Main(), _stBeamProcInfo::pidProc, sig_handler_m(), sig_handler_s(), and vProcInfo.

8.22.2 Variable Documentation

8.22.2.1 volatile bool bRun = true

Definition at line 44 of file BeamSrv.cc.

Referenced by clBeamSrvMaster::Abort(), clBeamSrv2Master::InDataThread(), clBeamSrvMaster::InDataThread(), main(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clBeamSrv2Master::ProcessLoop(), clBeamSrvMaster::ProcessLoop(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), clBeamSrv2Master::ServerThread(), clBeamSrvMaster::ServerThread(), sig_handler(), sig_handler_m(), SigHandler(), clSoundProxy::SoundInThread(), and clUIServer::Wait().

8.22.2.2 clDynThreads<clBeamSrvMaster>* BeamSrvMasterThreads

Definition at line 45 of file BeamSrv.cc.

Referenced by clBeamSrvMaster::InDataThread(), clBeamSrv2Master::Main(), clBeamSrvMaster::Main(), main(), clBeamSrvMaster::ServeClientThread(), clBeamSrv2Master::ServerThread(), and clBeamSrvMaster::ServerThread().

8.22.2.3 BS_PROCINFOV_T vProcInfo

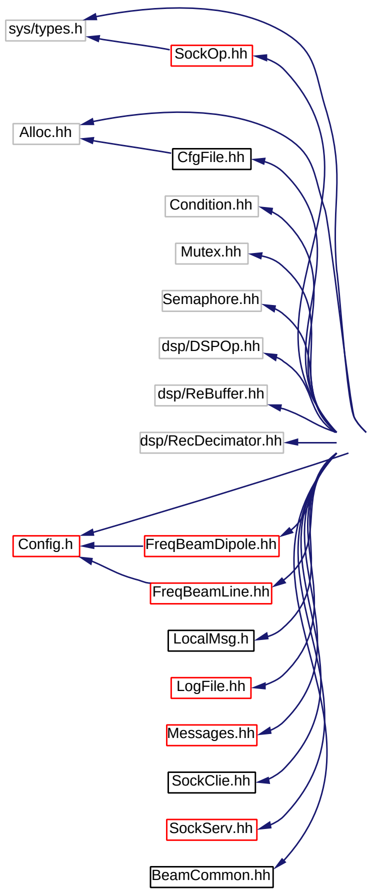
Definition at line 46 of file BeamSrv.cc.

Referenced by clBeamSrvMaster::InitProcessing(), main(), clBeamSrvMaster::ProcessLoop(), clBeamSrvMaster::SendInData(), clBeamSrvMaster::SendNodeParams(), and clBeamSrvMaster::WaitReady().

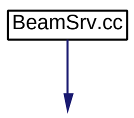
8.23 BeamSrv.hh File Reference

```
#include <sys/types.h>
#include <Alloc.hh>
#include <Condition.hh>
#include <Mutex.hh>
#include <Semaphore.hh>
#include <dsp/DSPOp.hh>
#include <dsp/ReBuffer.hh>
#include <dsp/RecDecimator.hh>
#include "Config.h"
#include "LocalMsg.h"
#include "CfgFile.hh"
#include "FreqBeamDipole.hh"
#include "FreqBeamLine.hh"
#include "LogFile.hh"
#include "Messages.hh"
#include "SockClie.hh"
#include "SockOp.hh"
#include "SockServ.hh"
#include "BeamCommon.hh"
```

Include dependency graph for BeamSrv.hh:



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



Compounds

- struct **_stBeamProcInfo**
- class **clBeamSrvMaster**

  Beamforming input server, main process.

- class **clBeamSrvSlave**

  Beamforming input server, slave process.

Defines

- #define **BS_PROCINFOV_T** std::vector<**stBeamProcInfo**>

Typedefs

- typedef **_stBeamProcInfo** **stBeamProcInfo**
- typedef **_stBeamProcInfo** * **stpBeamProcInfo**

8.23.1 Define Documentation

8.23.1.1 #define BS_PROCINFOV_T std::vector<stBeamProcInfo>

Definition at line 51 of file BeamSrv.hh.

8.23.2 Typedef Documentation

8.23.2.1 typedef struct _stBeamProcInfo stBeamProcInfo

8.23.2.2 typedef struct _stBeamProcInfo * stpBeamProcInfo

8.24 BeamSrv2.cc File Reference

`#include <stdio.h>`

`#include <stdlib.h>`

`#include <signal.h>`

`#include <limits.h>`

`#include <math.h>`

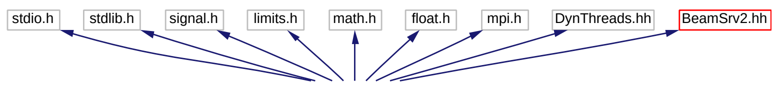
`#include <float.h>`

`#include <mpi.h>`

`#include <DynThreads.hh>`

`#include "BeamSrv2.hh"`

Include dependency graph for BeamSrv2.cc:



Functions

- void **sig_handler** (int signo)
- int **main** (int argc, char *argv[])
- bool **BeamCommNodeParams** (**stBeamNodeParams** &sNodeParams)

  Communicates beamformer parameters between nodes.

- bool **BeamCommInData** (GDT *fpData, int iDataCount)

Variables

- volatile bool **bRun** = true
- clDynThreads< **clBeamSrv2Master** > * **BeamSrvMasterThreads**

8.24.1 Function Documentation

8.24.1.1 void sig_handler (int *signo*)

Definition at line 41 of file BeamSrv2.cc.

References bRun.

Referenced by main().

8.24.1.2 int main (int *argc*, char * *argv*[ ])

Definition at line 55 of file BeamSrv2.cc.

References BeamSrvMasterThreads, GLOBAL_VERSMAJ, GLOBAL_-VERSMIN, GLOBAL_VERSPL, clBeamSrv2Slave::Main(), and sig_handler().

8.24.1.3 bool BeamCommNodeParams (stBeamNodeParams & *sNodeParams*)

Communicates beamformer parameters between nodes.

Definition at line 118 of file BeamSrv2.cc.

References _stBeamNodeParams::fSampleRate, _stBeamNodeParams::fSound-Speed, _stBeamNodeParams::fSpacing, _stBeamNodeParams::iBeamCount, _-stBeamNodeParams::iBlockSize, _stBeamNodeParams::iSensors, _stBeamNode-Params::iType, and _stBeamNodeParams::iWindowSize.

Referenced by clBeamSrv2Master::InitProcessing(), and clBeam-Srv2Slave::Main().

8.24.1.4 bool BeamCommInData (GDT * *fpData*, int *iDataCount*)

Definition at line 140 of file BeamSrv2.cc.

References GDT.

Referenced by clBeamSrv2Slave::Main(), and clBeamSrv2Master::Process-Loop().

8.24.2 Variable Documentation

8.24.2.1 volatile bool bRun = true

Definition at line 37 of file BeamSrv2.cc.

8.24.2.2 clDynThreads<clBeamSrv2Master>* BeamSrvMasterThreads

Definition at line 38 of file BeamSrv2.cc.

8.25 BeamSrv2.hh File Reference

#include <Alloc.hh>

#include <Condition.hh>

#include <Mutex.hh>

#include <Semaphore.hh>

#include <dsp/DSPOp.hh>

#include <dsp/RecDecimator.hh>

#include "Config.h"

#include "LocalMsg.h"

#include "CfgFile.hh"

#include "FreqBeamDipole.hh"

#include "FreqBeamLine.hh"

#include "LogFile.hh"

#include "Messages.hh"

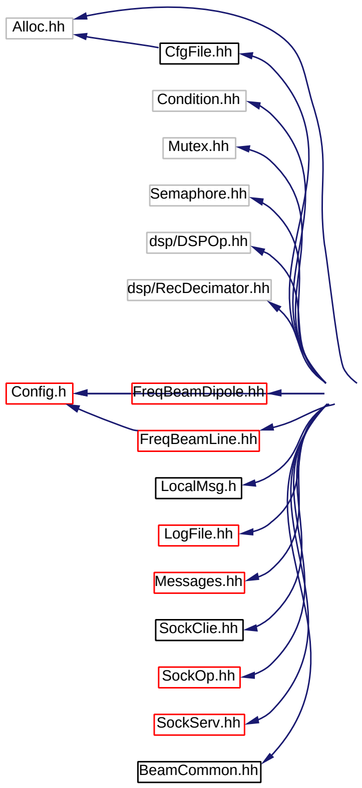
#include "SockClie.hh"

#include "SockOp.hh"

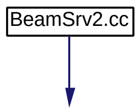
#include "SockServ.hh"

#include "BeamCommon.hh"

Include dependency graph for BeamSrv2.hh:



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



Compounds

- class **clBeamSrv2Master**

  Beamforming input server, root node.

- class **clBeamSrv2Slave**

  Beamforming input server, slave node.

Defines

- #define **BS_TAG_READY** 1

- #define **BS_TAG_RES** 2

Functions

- bool **BeamCommNodeParams** (**stBeamNodeParams** &)
  Communicates beamformer parameters between nodes.
- bool **BeamCommInData** (GDT *, int)

8.25.1 Define Documentation

8.25.1.1 #define BS_TAG_READY 1

Definition at line 47 of file BeamSrv2.hh.

Referenced by clBeamSrv2Slave::SendReady(), and clBeamSrv2Master::WaitReady().

8.25.1.2 #define BS_TAG_RES 2

Definition at line 48 of file BeamSrv2.hh.

Referenced by clBeamSrv2Master::ProcessLoop(), and clBeamSrv2Slave::SendRes().

8.25.2 Function Documentation

8.25.2.1 bool BeamCommNodeParams (stBeamNodeParams &)

Communicates beamformer parameters between nodes.

Definition at line 118 of file BeamSrv2.cc.

References _stBeamNodeParams::fSampleRate, _stBeamNodeParams::fSoundSpeed, _stBeamNodeParams::fSpacing, _stBeamNodeParams::iBeamCount, _stBeamNodeParams::iBlockSize, _stBeamNodeParams::iSensors, _stBeamNodeParams::iType, and _stBeamNodeParams::iWindowSize.

Referenced by clBeamSrv2Master::InitProcessing(), and clBeamSrv2Slave::Main().

8.25.2.2 bool BeamCommInData (GDT *, int)

Definition at line 140 of file BeamSrv2.cc.

References GDT.

Referenced by clBeamSrv2Slave::Main(), and clBeamSrv2Master::ProcessLoop().

8.26 CfgFile.cc File Reference

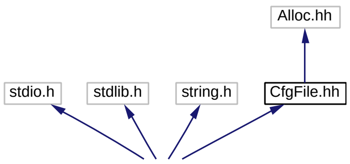
`#include <stdio.h>`

`#include <stdlib.h>`

`#include <string.h>`

`#include "CfgFile.hh"`

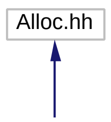
Include dependency graph for CfgFile.cc:



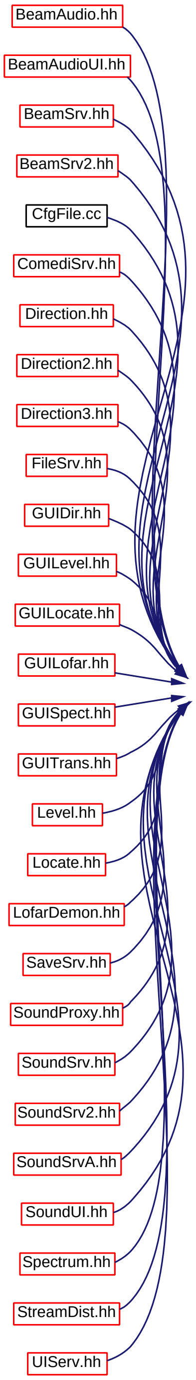
8.27 CfgFile.hh File Reference

#include <Alloc.hh>

Include dependency graph for CfgFile.hh:



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



Compounds

- class **clCfgFile**

Defines

- #define **CFGF_MAX_ENTRIES** 1024
- #define **CFGF_BUFSIZE** 1024

8.27.1 Define Documentation

8.27.1.1 #define CFGF_MAX_ENTRIES 1024

Definition at line 30 of file CfgFile.hh.

Referenced by clCfgFile::ReadFile().

8.27.1.2 #define CFGF_BUFSIZE 1024

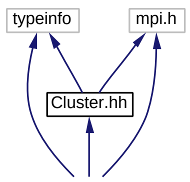
Definition at line 31 of file CfgFile.hh.

Referenced by clCfgFile::ReadFile().

8.28 Cluster.cc File Reference

```
#include <typeinfo>
#include <mpi.h>
#include "Cluster.hh"
```

Include dependency graph for Cluster.cc:

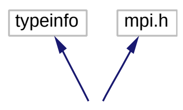


8.29 Cluster.hh File Reference

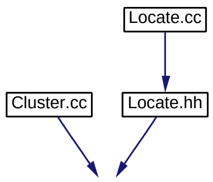
`#include <typeinfo>`

`#include <mpi.h>`

Include dependency graph for Cluster.hh:



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



Compounds

- class **clMPIProc**

  MPI process.

- class **clMPIComm**

  MPI communication.

8.30 ComediIO.cc File Reference

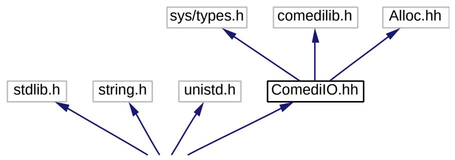
`#include <stdlib.h>`

`#include <string.h>`

`#include <unistd.h>`

`#include "ComediIO.hh"`

Include dependency graph for ComediIO.cc:



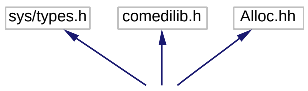
8.31 ComediIO.hh File Reference

`#include <sys/types.h>`

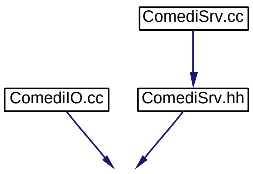
`#include <comedilib.h>`

`#include <Alloc.hh>`

Include dependency graph for ComediIO.hh:



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



Compounds

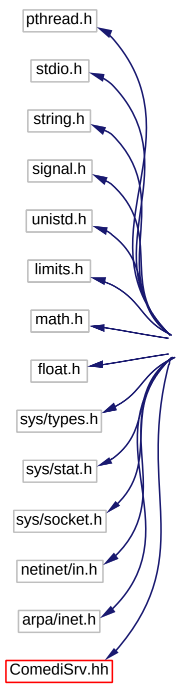
- class **clComediIO**

  Class for Comedi IO operations.

8.32 ComediSrv.cc File Reference

```
#include <pthread.h>
#include <stdio.h>
#include <string.h>
#include <signal.h>
#include <unistd.h>
#include <limits.h>
#include <math.h>
#include <float.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include "ComediSrv.hh"
```

Include dependency graph for ComediSrv.cc:



Functions

- int **main** (int argc, char *argv[ ])
- void **SigHandler** (int iSigNum)
- void * **WrapAudioInThread** (void *vpParam)
- void * **WrapServeClientThread** (void *vpParam)

Variables

- bool **bDaemon** = false
- **clComediSrv ComediSrv**

8.32.1 Function Documentation

8.32.1.1 int main (int *argc*, char * *argv*[ ])

Definition at line 44 of file ComediSrv.cc.

References ComediSrv, and clComediSrv::Main().

8.32.1.2 void SigHandler (int *iSigNum*)

Definition at line 50 of file ComediSrv.cc.

References clLogFile::Add(), ComediSrv, clComediSrv::Log, and clComediSrv::Stop().

8.32.1.3 void* WrapAudioInThread (void * *vpParam*)

Definition at line 73 of file ComediSrv.cc.

References clComediSrv::AudioInThread(), and ComediSrv.

Referenced by clStreamDist::Main(), clSoundSrvA::Main(), clSoundSrv2::Main(), and clComediSrv::Main().

8.32.1.4 void* WrapServeClientThread (void * *vpParam*)

Definition at line 79 of file ComediSrv.cc.

References ComediSrv, and clComediSrv::ServeClientThread().

Referenced by clStreamDist::Main(), clSoundSrvA::Main(), clSoundSrv2::Main(), clLocate::Main(), clComediSrv::Main(), and clSoundProxy::WaitConnectThread().

8.32.2 Variable Documentation

8.32.2.1 bool bDaemon = false [static]

Definition at line 40 of file ComediSrv.cc.

Referenced by clComediSrv::Main().

8.32.2.2 clComediSrv ComediSrv

Definition at line 41 of file ComediSrv.cc.

Referenced by main(), SigHandler(), WrapAudioInThread(), and WrapServeClientThread().

8.33 ComediSrv.hh File Reference

`#include <Alloc.hh>`

`#include <Condition.hh>`

`#include <Mutex.hh>`

`#include <dsp/DSPOp.hh>`

`#include <comedilib.h>`

`#include "Config.h"`

`#include "CfgFile.hh"`

`#include "ComediIO.hh"`

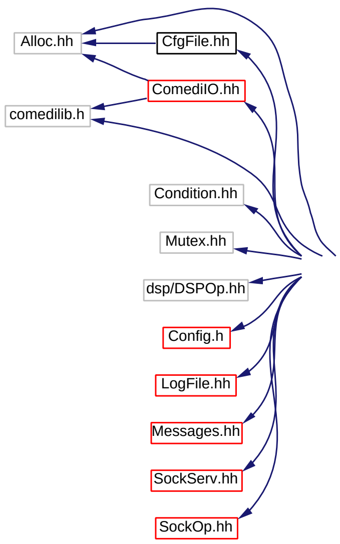
`#include "LogFile.hh"`

`#include "Messages.hh"`

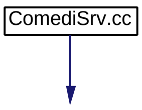
`#include "SockServ.hh"`

`#include "SockOp.hh"`

Include dependency graph for ComediSrv.hh:



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



Compounds

- class **clComediSrv**

  ComediServer.

Defines

- #define **COM_LOGENTRY_SIZE** 4096

8.33.1 Define Documentation

8.33.1.1 #define COM_LOGENTRY_SIZE 4096

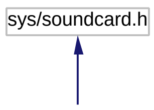
Definition at line 48 of file ComediSrv.hh.

Referenced by clComediSrv::AudioInThread(), and clComediSrv::ServeClientThread().

8.34 Config.h File Reference

`#include "sys/soundcard.h"`

Include dependency graph for Config.h:



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

Defines

- #define **CONSTF**
- #define **GLOBAL_DATATYPE** float
- #define **GDT** GLOBAL_DATATYPE
- #define **GCDT** stSCplx
- #define **GPDT** stSPolar
- #define **GUDT** utSCoord
- #define **GDT_SCAN** "%f"
- #define **GLOBAL_HEADER_LEN** 256
- #define **GLOBAL_SOCKET_PATH** "/tmp"
- #define **GLOBAL_VERSMAJ** 1
- #define **GLOBAL_VERSMIN** 7
- #define **GLOBAL_VERSPL** 0
- #define **SS_VERSMAJ** GLOBAL_VERSMAJ
- #define **SS_VERSMIN** GLOBAL_VERSMIN
- #define **SS_VERSPL** GLOBAL_VERSPL
- #define **SS_SND_FORMAT** AFMT_S16_LE
- #define **SS_SND_FORMAT_SIZE** 2
- #define **SS_SND_SAMPLERATE** 44100
- #define **SS_SND_CHANNELS** 2
- #define **SS_SND_DEVICE** "dsp0"
- #define **SS_LOGFILE** "log/soundsrv.log"
- #define **SS_SHUTDOWNFILE** "/tmp/soundsrv.shutdown"
- #define **SS_MAXCLIENTS** 8
- #define **SS_TIMEOUT** 250
- #define **SS_DEFAULT_PORT** 30000
- #define **SS_MAXERRORS** 100
- #define **SS_SCHED_PRIORITY** 1
- #define **SS_SOCKET_BUF_FRAGS** 16
- #define **SS2_CFGFILE** "soundsrv2.cfg"
- #define **SS2_LOGFILE** "log/soundsrv2.log"

- #define **SS2_SHUTDOWNFILE** "/tmp/soundsrv2.shutdown"
- #define **SS2_SOCKET_BUF_FRAGS** 16
- #define **SS2_FRAG_SIZE_DEFAULT** 4096
- #define **SS2_CONNECT_TIMEOUT** 250
- #define **SS2_INTHREAD_PRIORITY** 4
- #define **SS2_OUTTHREAD_PRIORITY** 3
- #define **SSA_CFGFILE** "soundsrva.cfg"
- #define **SSA_LOGFILE** "log/soundsrva.log"
- #define **SSA_SHUTDOWNFILE** "/tmp/soundsrva.shutdown"
- #define **SSA_SOCKET_BUF_FRAGS** 16
- #define **SSA_FRAG_SIZE_DEFAULT** 4096
- #define **SSA_CONNECT_TIMEOUT** 250
- #define **SSA_INTHREAD_PRIORITY** 4
- #define **SSA_OUTTHREAD_PRIORITY** 3
- #define **COM_CFGFILE** "comedisrv.cfg"
- #define **COM_LOGFILE** "log/comedisrv.log"
- #define **COM_SHUTDOWNFILE** "/tmp/comedisrv.shutdown"
- #define **COM_SOCKET_BUF_FRAGS** 16
- #define **COM_FRAG_SIZE_DEFAULT** 4096
- #define **COM_CONNECT_TIMEOUT** 250
- #define **COM_INTHREAD_PRIORITY** 4
- #define **COM_OUTTHREAD_PRIORITY** 3
- #define **COM_USE_DITHER** true
- #define **FS_CFGFILE** "filesrv.cfg"
- #define **FS_FRAG_SIZE_DEFAULT** 4096
- #define **FS_DEFAULT_PORT** 30000
- #define **FS_INTHREAD_PRIORITY** 4
- #define **FS_OUTTHREAD_PRIORITY** 3
- #define **BS_CFGFILE** "beamsrv.cfg"
- #define **BS_LOGFILE** "log/beamsrv.log"
- #define **BS_INTHREAD_PRIORITY** 4
- #define **BS_OUTTHREAD_PRIORITY** 3
- #define **SD_VERSMAJ** GLOBAL_VERSMAJ
- #define **SD_VERSMIN** GLOBAL_VERSMIN
- #define **SD_VERSPL** GLOBAL_VERSPL
- #define **SD_MAXCLIENTS** 128
- #define **SD_CONNECT_TIMEOUT** 250
- #define **SD_TIMEOUT** 1000
- #define **SD_CFGFILE** "streamdist.cfg"
- #define **SD_LOGFILE** "log/streamdist.log"
- #define **SD_SHUTDOWNFILE** "/tmp/streamdist.shutdown"
- #define **SD_MAX_ADDR_LEN** 128
- #define **SD_SCHED_PRIORITY** 1
- #define **SD_BUFFER_SIZE** 4096
- #define **SD_INTHREAD_PRIORITY** 2
- #define **SD_OUTTHREAD_PRIORITY** 1

- #define **SAVS_VERSMAJ** GLOBAL_VERSMAJ
- #define **SAVS_VERSMIN** GLOBAL_VERSMIN
- #define **SAVS_VERSPL** GLOBAL_VERSPL
- #define **SAVS_CFGFILE** "savesrv.cfg"
- #define **SAVS_LOGFILE** "log/savesrv.log"
- #define **SAVS_SHUTDOWNFILE** "/tmp/savesrv.shutdown"
- #define **SAVS_TIMEOUT** 1000
- #define **UIS_VERSMAJ** GLOBAL_VERSMAJ
- #define **UIS_VERSMIN** GLOBAL_VERSMIN
- #define **UIS_VERSPL** GLOBAL_VERSPL
- #define **UIS_CFGFILE** "uiserv.cfg"
- #define **UIS_LOGFILE** "log/uiserv.log"
- #define **UIS_SHUTDOWNFILE** "/tmp/uiserv.shutdown"
- #define **UIS_DEFAULT_PORT** 30001
- #define **UIS_TIMEOUT** 250
- #define **UIS_MSG_TIMEOUT** 1000
- #define **SPECT_CFGFILE** "spectrum.cfg"
- #define **SPECT_TIMEOUT** 250
- #define **SPECT_RAW1ST_TIMEOUT** 1000
- #define **SPECT_REQ_TIMEOUT** 5000
- #define **SPECT_DEF_FILTSIZE** 4096
- #define **SPECT_BAND_LIMIT** 10.0F
- #define **SGUI_CFGFILE** "guispect.cfg"
- #define **SGUI_HOSTFILE** "guispect.hosts"
- #define **SGUI_REQ_PROC** "spectrum"
- #define **AB_KBWIN_ALPHA** 3.0
- #define **BF_MAX_X_SENSORS** 64
- #define **BF_MAX_Y_SENSORS** 16
- #define **DIR_CFGFILE** "direction.cfg"
- #define **DIR_MAX_CPUS** 8
- #define **DIR_REQ_TIMEOUT** 5000
- #define **DIR_RAW1ST_TIMEOUT** 1000
- #define **DIR_TIMEOUT** 250
- #define **DIR_DB_SCALE** 144.4943979
- #define **DIR_DEF_WIN_SIZE** 2048
- #define **DIR2_CFGFILE** "direction2.cfg"
- #define **DIR2_REQ_TIMEOUT** 5000
- #define **DIR2_RAW1ST_TIMEOUT** 1000
- #define **DIR2_TIMEOUT** 250
- #define **DIR2_DEF_FILT_SIZE** 2048
- #define **DIR2_DEF_FFT_SIZE** 4096
- #define **DIR3_CFGFILE** "direction3.cfg"
- #define **DIR3_REQ_TIMEOUT** 5000
- #define **DIR3_RAW1ST_TIMEOUT** 1000
- #define **DIR3_TIMEOUT** 250
- #define **DIR3_DEF_FILT_SIZE** 2048

- #define **DGUI_CFGFILE** "guidir.cfg"
- #define **DGUI_HOSTFILE** "guidir.hosts"
- #define **DGUI_REQ_PROC** "direction"
- #define **DGUI_REQ_PROC2** "direction2"
- #define **DGUI_DEF_LINES** 300
- #define **DGUI_DEF_SOUNDSPEED** 1430.0
- #define **LOFAR_CFGFILE** "lofardemon.cfg"
- #define **LOFAR_REQ_TIMEOUT** 5000
- #define **LOFAR_RAW1ST_TIMEOUT** 1000
- #define **LOFAR_TIMEOUT** 250
- #define **LOFAR_DEF_FILTER_SIZE** 4096
- #define **LOFAR_DEF_DC_BLOCK** 1
- #define **LGUI_CFGFILE** "guilofar.cfg"
- #define **LGUI_HOSTFILE** "guilofar.hosts"
- #define **LGUI_REQ_PROC** "lofardemon"
- #define **LGUI_DEF_WIN_LENGTH** 1024
- #define **LGUI_DEF_LOW_FREQ** 0.0
- #define **LGUI_DEF_HIGH_FREQ** 1000.0
- #define **LGUI_DEF_REMOVE_NOISE** 0
- #define **LGUI_DEF_ALPHA** 1.5
- #define **LGUI_DEF_MEAN_LENGTH** 100
- #define **LGUI_DEF_GAP_LENGTH** 3
- #define **LGUI_DEF_HEIGHT** 100
- #define **SUI_MAX_CHANNELS** 8
- #define **SUI_SND_FORMAT** AFMT_S16_LE
- #define **SUI_SND_BITS** 16
- #define **SUI_SND_DATATYPE** signed short
- #define **SUI_SND_QUEUESIZE** 4096
- #define **SUI_SAMPLECOUNT** 2048
- #define **SUI_DEF_DEVICE** "/dev/dsp"
- #define **SUI_DEF_CHANNELS** 2
- #define **SUI_DEF_SAMPLERATE** 44100
- #define **SUI_CFGFILE** "soundui.cfg"
- #define **SUI_HOSTFILE** "soundui.hosts"
- #define **SUI_FIRST_TIMEOUT** 1000
- #define **SUI_IN_TIMEOUT** 250
- #define **SUI_VU_TIMEOUT** 200
- #define **SP_VERSMAJ** GLOBAL_VERSMAJ
- #define **SP_VERSMIN** GLOBAL_VERSMIN
- #define **SP_VERSPL** GLOBAL_VERSPL
- #define **SP_CFGFILE** "soundproxy.cfg"
- #define **SP_DEF_LOGFILE** "log/soundproxy.cfg"
- #define **SP_SERV_MAXLEN** 255
- #define **SP_MAXCLIENTS** 255
- #define **SP_1ST_MSG_TIMEOUT** 1000
- #define **SP_MSG_TIMEOUT** 250

- #define **SP_WAIT_CONN_TIMEOUT** 250
- #define **SP_SCHED_PRIORITY** 1
- #define **SP_BUFFER_SIZE** 4096
- #define **BA_CFGFILE** "beamaudio.cfg"
- #define **BA_1STREQ_TIMEOUT** 5000
- #define **BA_RAW1ST_TIMEOUT** 1000
- #define **BA_TIMEOUT** 250
- #define **BA_DEF_FRAGMENT_SIZE** 4096
- #define **BA_SCHED_PRIORITY** 2
- #define **BAUI_CFGFILE** "beamaudioui.cfg"
- #define **BAUI_HOSTFILE** "beamaudioui.hosts"
- #define **BAUI_REQ_PROC** "beamaudio"
- #define **BAUI_1ST_TIMEOUT** 5000
- #define **BAUI_DEF_SOUNDSPEED** "1430.0"
- #define **BAUI_DEF_DIR_RANGE** 180.0
- #define **BAUI_INT_DATATYPE** int
- #define **BAUI_SND_FORMAT** AFMT_S16_LE
- #define **BAUI_SND_BITS** 16
- #define **BAUI_SND_DATATYPE** signed short
- #define **LEVEL_CFGFILE** "level.cfg"
- #define **LEVEL_1STREQ_TIMEOUT** 5000
- #define **LEVEL_RAW1ST_TIMEOUT** 1000
- #define **LEVEL_TIMEOUT** 250
- #define **LEVEL_DEF_FILTER_SIZE** 4096
- #define **GUILEV_CFGFILE** "guilevel.cfg"
- #define **GUILEV_HOSTFILE** "guilevel.hosts"
- #define **GUILEV_REQ_PROC** "level"
- #define **LOCATE_CFGFILE** "locate.cfg"
- #define **LOCATE_LOGFILE** "log/locate.log"
- #define **LOCATE_SENSOR_LIST** "locate.sensors"
- #define **LOCATE_DIR_PROC** "direction3"
- #define **LOCATE_TIMEOUT** 250
- #define **LOCATE_DEF_WINDOWSIZE** 4096
- #define **LOCATE_DEF_SOUNDSPEED** 1430.0
- #define **LOCATE_DEF_LOWFREQ** 10.0
- #define **LOCATE_DEF_INTTIME** 1.0
- #define **LOCATE_DEF_SCALING** 0
- #define **LOCATE_DEF_SCALINGEXP** 2.0
- #define **LOCATE_DEF_REMOVENOISE** 0
- #define **LOCATE_DEF_ALPHA** 2.0
- #define **LOCATE_DEF_MEANLENGTH** 10
- #define **LOCATE_DEF_GAPLENGTH** 3
- #define **GUILOC_CFGFILE** "guilocate.cfg"
- #define **GUILOC_HOSTFILE** "guilocate.hosts"
- #define **GUITRANS_CFGFILE** "guitrans.cfg"
- #define **GUITRANS_HOSTFILE** "guitrans.hosts"

8.34.1 Define Documentation

8.34.1.1 #define CONSTF

Definition at line 44 of file Config.h.

8.34.1.2 #define GLOBAL_DATATYPE float

Definition at line 48 of file Config.h.

8.34.1.3 #define GDT GLOBAL_DATATYPE

Definition at line 49 of file Config.h.

Referenced by clCorrDipole::AddData(), clBeamDipole::AddData(), clArrayDipole::AddData(), clBeamAudioUI::AllocateBuffers(), clStreamDist::AudioInThread(), clSoundSrvA::AudioInThread(), clSoundSrv2::AudioInThread(), clComediSrv::AudioInThread(), BeamCommInData(), clDirection::CalcDipoleBeams(), clDirection::CalcDipoleCorrs(), clLofarDemon::CalcSpect(), clSpectDirLine2::Calculate(), clSpectDirLine::Calculate(), clSpectDirDipole2::Calculate(), clSpectDirDipole::Calculate(), clBeamAudio::clBeamAudio(), clBeamDipole::clBeamDipole(), clCorrDipole::clCorrDipole(), clLocateSensor::Clear(), clSoundChGUI::clSoundChGUI(), clSoundSrv::clSoundSrv(), clSpectDirDipole::clSpectDirDipole(), clSpectDirDipole2::clSpectDirDipole2(), clSpectDirLine::clSpectDirLine(), clSpectDirLine2::clSpectDirLine2(), clAudio3D::CopyToLocal(), clSpectrum::CreateFilter(), clGUILocate::DisplayResults(), clSaveSrv::Dither(), clFrameBuf::DrawColumn(), clFrameBuf::DrawLine(), clSpectGUI::DrawSpectrogram(), clSpectGUI::DrawSpectrum(), clSoundUI::Exec(), clDirection::Exec(), clBeamSrv2Master::FilterData(), clBeamSrvMaster::FilterData(), clFreqBeamLine::Get(), clFreqBeamDipole::Get(), clArrayDipole::GetArrayFreq(), clSpectDir2::GetArrayFrequency(), clSpectDir::GetArrayFrequency(), clArrayDipole::GetAudioData(), clSpectrum::GetData(), clArrayDipole::GetDelay(), clArrayDipole::GetDelaySamples(), clArrayDipole::GetDelayTime(), clSpectDir2::GetDirection(), clSpectDir::GetDirection(), clSpectDirLine2::GetDirections(), clSpectDirDipole2::GetDirections(), clSpectDir::GetDirections(), clArrayDipole::GetFilteredData(), clArrayDipole::GetFilteredDataPtr(), clArrayDipole::GetFiltPtr(), clSpectDirLine2::GetFreqResolution(), clSpectDirDipole2::GetFreqResolution(), clArrayDipole::GetHFPtr(), clSpectDirLine2::GetIntegrationTime(), clSpectDirLine::GetIntegrationTime(), clSpectDirDipole2::GetIntegrationTime(), clSpectDirDipole::GetIntegrationTime(), clCorrDipole::GetIntegrationTime(), clBeamDipole::GetIntegrationTime(), clSpectDirLine2::GetLevels(), clSpectDirDipole2::GetLevels(), clArrayDipole::GetMaxDelay(), clCorrDipole::GetPeakLevel(), clBeamDipole::GetPeakLevel(), clArrayDipole::GetRawData(), clArrayDipole::GetRawDataPtr(), clArrayDipole::GetRawPtr(), clLocateSensor::GetResultMatrix(), clSpectDirLine2::GetResults(), clSpectDirLine::GetResults(), clSpectDirDipole2::GetResults(), clSpectDirDipole::GetResults(), clSpectGUI::GetSettings(), clBeamSrv2Master::InDataThread(), clBeamSrvMaster::InData-

Thread(), clXMMSOut::Init(), clSpectrum::Init(), clDirection::InitArray(), clBeamAudio::InitBeam(), clGUILofar::InitConnection(), clGUIDir::InitConnection(), clLofarDemon::InitFFT(), clLofarDemon::InitFilter(), clBeamSrv2Master::InitFilterBank(), clBeamSrvMaster::InitFilterBank(), clSaveSrv::Initialize(), clLocateSystem::Initialize(), clLocateSensor::Initialize(), clSubLocate::Initialize(), clLocate::Initialize(), clLevel::Initialize(), clFreqBeamLine::Initialize(), clAudio3D::Initialize(), clArraySensor::Initialize(), clArrayDipole::Initialize(), clSoundSrv::InputExec(), main(), clFileSrv::Main(), clBeamSrv2Slave::Main(), clBeamSrvSlave::Main(), clLofarDemon::MainLoop(), clSoundUI::OnApplyCurveClicked(), clGUILocate::OnConnectClick(), clGUILofar::OnExposeLine(), clSpectGUI::OnExposeSpect(), clGUILofar::OnGdkInput(), clGUILocate::OnGdkInput(), clGUIDir::OnGdkInput(), clSoundUI::OnValueChangedEvent(), clAudio3D::Prepare(), clLocateSystem::Process(), clLevel::Process(), clCorrDipole::Process(), clBeamDipole::Process(), clAudio3D::Process(), clAudio3D::ProcessDistance(), clSpectrum::ProcessLoop(), clDirection3::ProcessLoop(), clDirection2::ProcessLoop(), clBeamSrv2Master::ProcessLoop(), clBeamSrvMaster::ProcessLoop(), clBeamAudio::ProcessLoop(), clLocate::ProcessThread(), clSpectDirLine2::PutData(), clSpectDirLine::PutData(), clSpectDirDipole2::PutData(), clSpectDirDipole::PutData(), clFileSrv::ReaderThread(), clLevel::ReceiveData(), clBeamSrvSlave::RecvInData(), clDirection::Scale(), clBeamSrvMaster::SendInData(), clBeamSrv2Slave::SendRes(), clBeamSrvSlave::SendRes(), clBeamAudio::SendResult(), clLofarDemon::SendResults(), clDirection::SendResults(), clXMMSOut::ServeClientThread(), clFileSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), clSpectDir::SetDirection(), clLocateSensor::SetDirection(), clArraySensor::SetDirection(), clSpectDir::SetDirectionCount(), clLocateSensor::SetDirectionValues(), clCorrDipole::SetHistory(), clBeamDipole::SetHistory(), clArrayDipole::SetLowFreqLimit(), clArraySensor::SetSampleRate(), clArrayBase::SetSampleRate(), clArrayBase::SetShading(), clArraySensor::SetSoundSpeed(), clArrayBase::SetSoundSpeed(), clSoundUI::SoundInThread(), clSoundUI::SoundOutThread(), clLofarDemon::SpectStdDev(), clXMMSOut::WriteAudio(), and clSaveSrv::WriteData().

8.34.1.4 #define GCDT stSCplx

Definition at line 50 of file Config.h.

Referenced by clLofarDemon::CalcSpect(), clSpectDirLine2::Calculate(), clSpectDirLine::Calculate(), clSpectDirDipole2::Calculate(), clSpectDirDipole::Calculate(), clSpectDirDipole::clSpectDirDipole(), clSpectDirDipole2::clSpectDirDipole2(), clSpectDirLine::clSpectDirLine(), clSpectDirLine2::clSpectDirLine2(), clAudio3D::Initialize(), and clSpectrum::ProcessLoop().

8.34.1.5 #define GPDT stSPolar

Definition at line 51 of file Config.h.

8.34.1.6 #define GUDT utSCoord

Definition at line 52 of file Config.h.

8.34.1.7 #define GDT_SCAN "%f"

Definition at line 53 of file Config.h.

8.34.1.8 #define GLOBAL_HEADER_LEN 256

Definition at line 54 of file Config.h.

Referenced by clBeamAudioUI::AllocateBuffers(), clStreamDist::AudioInThread(), clSoundProxy::clSoundProxy(), clSubLocate::ConnectDir(), clGUILocate::ConnectToServer(), clGUILevel::ConnectToServer(), clSoundMsg::GetData(), clLevel::GetRequest(), clDirection::GetRequest(), clLofarDemon::GetRequestMsg(), clLocateMsg::GetResult(), clBeamAudioMsg::GetResult(), clLofarMsg::GetResult(), clDirMsg2::GetResult(), clDirMsg::GetResult(), clSpectMsg::GetResult(), clSpectrum::GetRq(), clDirection3::GetRq(), clDirection2::GetRq(), clBeamAudio::GetRq(), clSpectGUI::GetSettings(), clSpectrum::Init(), clSpectGUI::InitConnection(), clGUILofar::InitConnection(), clGUIDir::InitConnection(), clBeamAudioUI::InitConnection(), clSubLocate::Initialize(), clLocate::Initialize(), main(), clGUILocate::OnConnectClick(), clGUILofar::OnGdkInput(), clGUILevel::OnGdkInput(), clDirection3::ProcessLoop(), clDirection2::ProcessLoop(), clBeamAudio::SendFirst(), clLevel::SendResult(), clBeamAudio::SendResult(), clLofarDemon::SendResults(), clDirection::SendResults(), clSpectGUI::SendSettings(), clGUILofar::SendSettings(), clGUILevel::SendSettings(), clGUIDir::SendSettings(), clBeamAudioUI::SendSettings(), clSoundSrv::ServeClientExec(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clSoundProxy::ServeClientThread(), clLocate::ServeClientThread(), clFileSrv::ServeClientThread(), clComediSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), clSoundMsg::SetData(), clBeamAudioMsg::SetFirst(), clLocateMsg::SetHeader(), clLofarMsg::SetRequest(), clSpectMsg::SetRequest(), clLocateMsg::SetResult(), clLevelMsg::SetResult(), clBeamAudioMsg::SetResult(), clLofarMsg::SetResult(), clDirMsg2::SetResult(), clDirMsg::SetResult(), clSpectMsg::SetResult(), clSoundMsg::SetStart(), clSoundUI::SoundInThread(), clSoundProxy::SoundInThread(), and clBeamAudioUI::UpdateSettings().

8.34.1.9 #define GLOBAL_SOCKET_PATH "/tmp"

Definition at line 55 of file Config.h.

Referenced by clSockServ::Bind(), and clSockClie::Connect().

8.34.1.10 #define GLOBAL_VERSMAJ 1

Definition at line 56 of file Config.h.

Referenced by clXMMSOut::About(), get_oplugin_info(), clStreamDist::Main(), clSoundSrvA::Main(), clSoundSrv2::Main(), clComediSrv::Main(), main(), and clFrameBuf::SetTiffTags().

8.34.1.11 #define GLOBAL_VERSMIN 7

Definition at line 57 of file Config.h.

Referenced by clXMMSOut::About(), get_oplugin_info(), clStreamDist::Main(), clSoundSrvA::Main(), clSoundSrv2::Main(), clComediSrv::Main(), main(), and clFrameBuf::SetTiffTags().

8.34.1.12 #define GLOBAL_VERSPL 0

Definition at line 58 of file Config.h.

Referenced by clXMMSOut::About(), get_oplugin_info(), clStreamDist::Main(), clSoundSrvA::Main(), clSoundSrv2::Main(), clComediSrv::Main(), main(), and clFrameBuf::SetTiffTags().

8.34.1.13 #define SS_VERSMAJ GLOBAL_VERSMAJ

Definition at line 61 of file Config.h.

Referenced by main().

8.34.1.14 #define SS_VERSMIN GLOBAL_VERSMIN

Definition at line 62 of file Config.h.

Referenced by main().

8.34.1.15 #define SS_VERSPL GLOBAL_VERSPL

Definition at line 63 of file Config.h.

Referenced by main().

8.34.1.16 #define SS_SND_FORMAT AFMT_S16_LE

Definition at line 64 of file Config.h.

Referenced by clSoundSrv::clSoundSrv().

8.34.1.17 #define SS_SND_FORMAT_SIZE 2

Definition at line 65 of file Config.h.

Referenced by clSoundSrv::clSoundSrv().

8.34.1.18 #define SS_SND_SAMPLERATE 44100

Definition at line 66 of file Config.h.

Referenced by clSoundSrv::clSoundSrv().

8.34.1.19 #define SS_SND_CHANNELS 2

Definition at line 67 of file Config.h.

Referenced by clSoundSrv::clSoundSrv().

8.34.1.20 #define SS_SND_DEVICE "dsp0"

Definition at line 68 of file Config.h.

Referenced by main().

8.34.1.21 #define SS_LOGFILE "log/soundsrv.log"

Definition at line 69 of file Config.h.

Referenced by clSoundSrv::clSoundSrv().

8.34.1.22 #define SS_SHUTDOWNFILE "/tmp/soundsrv.shutdown"

Definition at line 70 of file Config.h.

Referenced by clSoundSrv::InputExec().

8.34.1.23 #define SS_MAXCLIENTS 8

Definition at line 71 of file Config.h.

Referenced by clSoundSrv::clSoundSrv(), clSoundSrv::FindFreeClient(), clSoundSrv::FindThisHandle(), and clSoundSrv::WaitConnectExec().

8.34.1.24 #define SS_TIMEOUT 250

Definition at line 72 of file Config.h.

8.34.1.25 #define SS_DEFAULT_PORT 30000

Definition at line 73 of file Config.h.

Referenced by clSoundSrv::WaitConnectExec().

8.34.1.26 #define SS_MAXERRORS 100

Definition at line 74 of file Config.h.

Referenced by clSoundSrv::InputExec().

8.34.1.27 #define SS_SCHED_PRIORITY 1

Definition at line 75 of file Config.h.

Referenced by main().

8.34.1.28 #define SS_SOCKET_BUF_FRAGS 16

Definition at line 76 of file Config.h.

Referenced by clSoundSrv::ServeClientExec().

8.34.1.29 #define SS2_CFGFILE "soundsrv2.cfg"

Definition at line 79 of file Config.h.

Referenced by clSoundSrv2::clSoundSrv2().

8.34.1.30 #define SS2_LOGFILE "log/soundsrv2.log"

Definition at line 80 of file Config.h.

Referenced by clSoundSrv2::clSoundSrv2().

8.34.1.31 #define SS2_SHUTDOWNFILE "/tmp/soundsrv2.shutdown"

Definition at line 81 of file Config.h.

Referenced by clSoundSrv2::Main().

8.34.1.32 #define SS2_SOCKET_BUF_FRAGS 16

Definition at line 82 of file Config.h.

Referenced by clSoundSrv2::ServeClientThread().

8.34.1.33 #define SS2_FRAG_SIZE_DEFAULT 4096

Definition at line 83 of file Config.h.

Referenced by clSoundSrv2::AudioInThread().

8.34.1.34 #define SS2_CONNECT_TIMEOUT 250

Definition at line 84 of file Config.h.

Referenced by clSoundSrv2::Main().

8.34.1.35 #define SS2_INTHREAD_PRIORITY 4

Definition at line 85 of file Config.h.

Referenced by clSoundSrv2::AudioInThread().

8.34.1.36 #define SS2_OUTTHREAD_PRIORITY 3

Definition at line 86 of file Config.h.

Referenced by clSoundSrv2::ServeClientThread().

8.34.1.37 #define SSA_CFGFILE "soundsrva.cfg"

Definition at line 89 of file Config.h.

Referenced by clSoundSrvA::clSoundSrvA().

8.34.1.38 #define SSA_LOGFILE "log/soundsrva.log"

Definition at line 90 of file Config.h.

Referenced by clSoundSrvA::clSoundSrvA().

8.34.1.39 #define SSA_SHUTDOWNFILE "/tmp/soundsrva.shutdown"

Definition at line 91 of file Config.h.

Referenced by clSoundSrvA::Main().

8.34.1.40 #define SSA_SOCKET_BUF_FRAGS 16

Definition at line 92 of file Config.h.

Referenced by clSoundSrvA::ServeClientThread().

8.34.1.41 #define SSA_FRAG_SIZE_DEFAULT 4096

Definition at line 93 of file Config.h.

Referenced by clSoundSrvA::AudioInThread(), and clSoundSrvA::GetAudioCfg().

8.34.1.42 #define SSA_CONNECT_TIMEOUT 250

Definition at line 94 of file Config.h.

Referenced by clSoundSrvA::Main().

8.34.1.43 #define SSA_INTHREAD_PRIORITY 4

Definition at line 95 of file Config.h.

Referenced by clSoundSrvA::AudioInThread().

8.34.1.44 #define SSA_OUTTHREAD_PRIORITY 3

Definition at line 96 of file Config.h.

Referenced by clSoundSrvA::ServeClientThread().

8.34.1.45 #define COM_CFGFILE "comedisrv.cfg"

Definition at line 99 of file Config.h.

Referenced by clComediSrv::clComediSrv().

8.34.1.46 #define COM_LOGFILE "log/comedisrv.log"

Definition at line 100 of file Config.h.

Referenced by clComediSrv::clComediSrv().

8.34.1.47 #define COM_SHUTDOWNFILE "/tmp/comedisrv.shutdown"

Definition at line 101 of file Config.h.

Referenced by clComediSrv::Main().

8.34.1.48 #define COM_SOCKET_BUF_FRAGS 16

Definition at line 102 of file Config.h.

Referenced by clComediSrv::ServeClientThread().

8.34.1.49 #define COM_FRAG_SIZE_DEFAULT 4096

Definition at line 103 of file Config.h.

Referenced by clComediSrv::AudioInThread().

8.34.1.50 #define COM_CONNECT_TIMEOUT 250

Definition at line 104 of file Config.h.

Referenced by clComediSrv::Main().

8.34.1.51 #define COM_INTHREAD_PRIORITY 4

Definition at line 105 of file Config.h.

Referenced by clComediSrv::AudioInThread().

8.34.1.52 #define COM_OUTTHREAD_PRIORITY 3

Definition at line 106 of file Config.h.

Referenced by clComediSrv::ServeClientThread().

8.34.1.53 #define COM_USE_DITHER true

Definition at line 107 of file Config.h.

Referenced by clComediSrv::AudioInThread().

8.34.1.54 #define FS_CFGFILE "filesrv.cfg"

Definition at line 110 of file Config.h.

Referenced by clFileSrv::Main().

8.34.1.55 #define FS_FRAG_SIZE_DEFAULT 4096

Definition at line 111 of file Config.h.

Referenced by clFileSrv::Main().

8.34.1.56 #define FS_DEFAULT_PORT 30000

Definition at line 112 of file Config.h.

Referenced by clFileSrv::Main().

8.34.1.57 #define FS_INTHREAD_PRIORITY 4

Definition at line 113 of file Config.h.

Referenced by clFileSrv::ReaderThread().

8.34.1.58 #define FS_OUTTHREAD_PRIORITY 3

Definition at line 114 of file Config.h.

Referenced by clFileSrv::ServeClientThread().

8.34.1.59 #define BS_CFGFILE "beamsrv.cfg"

Definition at line 117 of file Config.h.

Referenced by clBeamSrv2Master::Main(), and clBeamSrvMaster::Main().

8.34.1.60 #define BS_LOGFILE "log/beamsrv.log"

Definition at line 118 of file Config.h.

Referenced by clBeamSrv2Master::clBeamSrv2Master(), and clBeamSrv-Master::clBeamSrvMaster().

8.34.1.61 #define BS_INTHREAD_PRIORITY 4

Definition at line 119 of file Config.h.

Referenced by clBeamSrvMaster::InDataThread().

8.34.1.62 #define BS_OUTTHREAD_PRIORITY 3

Definition at line 120 of file Config.h.

Referenced by clBeamSrvMaster::ServeClientThread().

8.34.1.63 #define SD_VERSMAJ GLOBAL_VERSMAJ

Definition at line 123 of file Config.h.

8.34.1.64 #define SD_VERSMIN GLOBAL_VERSMIN

Definition at line 124 of file Config.h.

8.34.1.65 #define SD_VERSPL GLOBAL_VERSPL

Definition at line 125 of file Config.h.

8.34.1.66 #define SD_MAXCLIENTS 128

Definition at line 126 of file Config.h.

8.34.1.67 #define SD_CONNECT_TIMEOUT 250

Definition at line 127 of file Config.h.

Referenced by clStreamDist::Main().

8.34.1.68 #define SD_TIMEOUT 1000

Definition at line 128 of file Config.h.

8.34.1.69 #define SD_CFGFILE "streamdist.cfg"

Definition at line 129 of file Config.h.

Referenced by clDirection::clDirection(), clStreamDist::clStreamDist(), clSpectrum::GetCfg(), clLevel::GetCfg(), clDirection3::GetCfg(), clDirection2::GetCfg(), clBeamAudio::GetCfg(), clLofarDemon::GetDataHdr(), and clSaveSrv::Initialize().

8.34.1.70 #define SD_LOGFILE "log/streamdist.log"

Definition at line 130 of file Config.h.

Referenced by clStreamDist::clStreamDist().

8.34.1.71 #define SD_SHUTDOWNFILE "/tmp/streamdist.shutdown"

Definition at line 131 of file Config.h.

Referenced by clStreamDist::Main().

8.34.1.72 #define SD_MAX_ADDR_LEN 128

Definition at line 132 of file Config.h.

8.34.1.73 #define SD_SCHED_PRIORITY 1

Definition at line 133 of file Config.h.

8.34.1.74 #define SD_BUFFER_SIZE 4096

Definition at line 134 of file Config.h.

8.34.1.75 #define SD_INTHREAD_PRIORITY 2

Definition at line 135 of file Config.h.

Referenced by clStreamDist::AudioInThread().

8.34.1.76 #define SD_OUTTHREAD_PRIORITY 1

Definition at line 136 of file Config.h.

Referenced by clStreamDist::ServeClientThread().

8.34.1.77 #define SAVS_VERSMAJ GLOBAL_VERSMAJ

Definition at line 139 of file Config.h.

Referenced by clSaveSrv::Main().

8.34.1.78 #define SAVS_VERSMIN GLOBAL_VERSMIN

Definition at line 140 of file Config.h.

Referenced by clSaveSrv::Main().

8.34.1.79 #define SAVS_VERSPL GLOBAL_VERSPL

Definition at line 141 of file Config.h.

Referenced by clSaveSrv::Main().

8.34.1.80 #define SAVS_CFGFILE "savesrv.cfg"

Definition at line 142 of file Config.h.

Referenced by clSaveSrv::Initialize().

8.34.1.81 #define SAVS_LOGFILE "log/savesrv.log"

Definition at line 143 of file Config.h.

Referenced by clSaveSrv::clSaveSrv().

8.34.1.82 #define SAVS_SHUTDOWNFILE "/tmp/savesrv.shutdown"

Definition at line 144 of file Config.h.

Referenced by clSaveSrv::ProcessLoop().

8.34.1.83 #define SAVS_TIMEOUT 1000

Definition at line 145 of file Config.h.

8.34.1.84 #define UIS_VERSMAJ GLOBAL_VERSMAJ

Definition at line 148 of file Config.h.

Referenced by main().

8.34.1.85 #define UIS_VERSMIN GLOBAL_VERSMIN

Definition at line 149 of file Config.h.

Referenced by main().

8.34.1.86 #define UIS_VERSPL GLOBAL_VERSPL

Definition at line 150 of file Config.h.

Referenced by main().

8.34.1.87 #define UIS_CFGFILE "uiserv.cfg"

Definition at line 151 of file Config.h.

Referenced by clUIServer::clUIServer().

8.34.1.88 #define UIS_LOGFILE "log/uiserv.log"

Definition at line 152 of file Config.h.

Referenced by clUIServer::clUIServer().

8.34.1.89 #define UIS_SHUTDOWNFILE "/tmp/uiserv.shutdown"

Definition at line 153 of file Config.h.

Referenced by clUIServer::Wait().

8.34.1.90 #define UIS_DEFAULT_PORT 30001

Definition at line 154 of file Config.h.

Referenced by clUIServer::clUIServer().

8.34.1.91 #define UIS_TIMEOUT 250

Definition at line 155 of file Config.h.

Referenced by clUIServer::Wait().

8.34.1.92 #define UIS_MSG_TIMEOUT 1000

Definition at line 156 of file Config.h.

Referenced by main().

8.34.1.93 #define SPECT_CFGFILE "spectrum.cfg"

Definition at line 159 of file Config.h.

Referenced by clSpectrum::GetCfg().

8.34.1.94 #define SPECT_TIMEOUT 250

Definition at line 160 of file Config.h.

8.34.1.95 #define SPECT_RAW1ST_TIMEOUT 1000

Definition at line 161 of file Config.h.

Referenced by clSpectrum::ConnectStream().

8.34.1.96 #define SPECT_REQ_TIMEOUT 5000

Definition at line 162 of file Config.h.

Referenced by clSpectrum::GetRq().

8.34.1.97 #define SPECT_DEF_FILTSIZE 4096

Definition at line 163 of file Config.h.

Referenced by clSpectrum::GetCfg().

8.34.1.98 #define SPECT_BAND_LIMIT 10.0F

Definition at line 166 of file Config.h.

Referenced by clSpectGUI::CountBands().

8.34.1.99 #define SGUI_CFGFILE "guispect.cfg"

Definition at line 170 of file Config.h.

Referenced by clSpectGUI::clSpectGUI().

8.34.1.100 #define SGUI_HOSTFILE "guispect.hosts"

Definition at line 171 of file Config.h.

Referenced by clSpectGUI::BuildTable1().

8.34.1.101 #define SGUI_REQ_PROC "spectrum"

Definition at line 172 of file Config.h.

Referenced by clSpectGUI::InitConnection().

8.34.1.102 #define AB_KBWIN_ALPHA 3.0

Definition at line 175 of file Config.h.

Referenced by clArrayBase::SetShading().

8.34.1.103 #define BF_MAX_X_SENSORS 64

Definition at line 176 of file Config.h.

8.34.1.104 #define BF_MAX_Y_SENSORS 16

Definition at line 177 of file Config.h.

8.34.1.105 #define DIR_CFGFILE "direction.cfg"

Definition at line 180 of file Config.h.

Referenced by clDirection::clDirection().

8.34.1.106 #define DIR_MAX_CPUS 8

Definition at line 181 of file Config.h.

Referenced by clDirection::clDirection().

8.34.1.107 #define DIR_REQ_TIMEOUT 5000

Definition at line 182 of file Config.h.

Referenced by clDirection::GetRequest().

8.34.1.108 #define DIR_RAW1ST_TIMEOUT 1000

Definition at line 183 of file Config.h.

Referenced by clDirection::GetFirstMsg().

8.34.1.109 #define DIR_TIMEOUT 250

Definition at line 184 of file Config.h.

Referenced by clDirection::ReceiveData(), and clDirection::SendResults().

8.34.1.110 #define DIR_DB_SCALE 144.4943979

Definition at line 185 of file Config.h.

Referenced by clCorrDipole::GetPeakLevel(), and clBeamDipole::GetPeakLevel().

8.34.1.111 #define DIR_DEF_WIN_SIZE 2048

Definition at line 186 of file Config.h.

Referenced by clDirection::clDirection().

8.34.1.112 #define DIR2_CFGFILE "direction2.cfg"

Definition at line 189 of file Config.h.

Referenced by clDirection2::GetCfg().

8.34.1.113 #define DIR2_REQ_TIMEOUT 5000

Definition at line 190 of file Config.h.

Referenced by clDirection2::GetRq().

8.34.1.114 #define DIR2_RAW1ST_TIMEOUT 1000

Definition at line 191 of file Config.h.

Referenced by clDirection2::ConnectStream().

8.34.1.115 #define DIR2_TIMEOUT 250

Definition at line 192 of file Config.h.

8.34.1.116 #define DIR2_DEF_FILT_SIZE 2048

Definition at line 193 of file Config.h.

Referenced by clDirection2::GetCfg().

8.34.1.117 #define DIR2_DEF_FFT_SIZE 4096

Definition at line 194 of file Config.h.

Referenced by clDirection2::GetCfg().

8.34.1.118 #define DIR3_CFGFILE "direction3.cfg"

Definition at line 197 of file Config.h.

Referenced by clDirection3::GetCfg().

8.34.1.119 #define DIR3_REQ_TIMEOUT 5000

Definition at line 198 of file Config.h.

Referenced by clDirection3::GetRq().

8.34.1.120 #define DIR3_RAW1ST_TIMEOUT 1000

Definition at line 199 of file Config.h.

Referenced by clDirection3::ConnectStream().

8.34.1.121 #define DIR3_TIMEOUT 250

Definition at line 200 of file Config.h.

8.34.1.122 #define DIR3_DEF_FILT_SIZE 2048

Definition at line 201 of file Config.h.

Referenced by clDirection3::GetCfg().

8.34.1.123 #define DGUI_CFGFILE "guidir.cfg"

Definition at line 205 of file Config.h.

Referenced by clGUIDir::clGUIDir().

8.34.1.124 #define DGUI_HOSTFILE "guidir.hosts"

Definition at line 206 of file Config.h.

Referenced by clGUIDir::BuildTable1().

8.34.1.125 #define DGUI_REQ_PROC "direction"

Definition at line 207 of file Config.h.

Referenced by clGUIDir::InitConnection().

8.34.1.126 #define DGUI_REQ_PROC2 "direction2"

Definition at line 208 of file Config.h.

Referenced by clGUIDir::InitConnection().

8.34.1.127 #define DGUI_DEF_LINES 300

Definition at line 209 of file Config.h.

Referenced by clGUIDir::Exec().

8.34.1.128 #define DGUI_DEF_SOUNDSPEED 1430.0

Definition at line 210 of file Config.h.

Referenced by clGUIDir::Exec().

8.34.1.129 #define LOFAR_CFGFILE "lofardemon.cfg"

Definition at line 213 of file Config.h.

Referenced by clLofarDemon::Initialize().

8.34.1.130 #define LOFAR_REQ_TIMEOUT 5000

Definition at line 214 of file Config.h.

Referenced by clLofarDemon::GetRequestMsg().

8.34.1.131 #define LOFAR_RAW1ST_TIMEOUT 1000

Definition at line 215 of file Config.h.

Referenced by clLofarDemon::GetDataHdr().

8.34.1.132 #define LOFAR_TIMEOUT 250

Definition at line 216 of file Config.h.

Referenced by clLofarDemon::SendResults().

8.34.1.133 #define LOFAR_DEF_FILTER_SIZE 4096

Definition at line 217 of file Config.h.

Referenced by clLofarDemon::InitFilter().

8.34.1.134 #define LOFAR_DEF_DC_BLOCK 1

Definition at line 218 of file Config.h.

Referenced by clLofarDemon::InitFilter().

8.34.1.135 #define LGUI_CFGFILE "guilofar.cfg"

Definition at line 221 of file Config.h.

Referenced by clGUILofar::clGUILofar().

8.34.1.136 #define LGUI_HOSTFILE "guilofar.hosts"

Definition at line 222 of file Config.h.

Referenced by clGUILofar::BuildTable1().

8.34.1.137 #define LGUI_REQ_PROC "lofardemon"

Definition at line 223 of file Config.h.

Referenced by clGUILofar::InitConnection().

8.34.1.138 #define LGUI_DEF_WIN_LENGTH 1024

Definition at line 224 of file Config.h.

Referenced by clGUILofar::clGUILofar().

8.34.1.139 #define LGUI_DEF_LOW_FREQ 0.0

Definition at line 225 of file Config.h.

Referenced by clGUILofar::clGUILofar().

8.34.1.140 #define LGUI_DEF_HIGH_FREQ 1000.0

Definition at line 226 of file Config.h.

Referenced by clGUILofar::clGUILofar().

8.34.1.141 #define LGUI_DEF_REMOVE_NOISE 0

Definition at line 227 of file Config.h.

Referenced by clGUILofar::clGUILofar().

8.34.1.142 #define LGUI_DEF_ALPHA 1.5

Definition at line 228 of file Config.h.

Referenced by clGUILofar::clGUILofar().

8.34.1.143 #define LGUI_DEF_MEAN_LENGTH 100

Definition at line 229 of file Config.h.

Referenced by clGUILofar::clGUILofar().

8.34.1.144 #define LGUI_DEF_GAP_LENGTH 3

Definition at line 230 of file Config.h.

Referenced by clGUILofar::clGUILofar().

8.34.1.145 #define LGUI_DEF_HEIGHT 100

Definition at line 231 of file Config.h.

Referenced by clGUILofar::clGUILofar().

8.34.1.146 #define SUI_MAX_CHANNELS 8

Definition at line 234 of file Config.h.

Referenced by clSoundUI::clSoundUI().

8.34.1.147 #define SUI_SND_FORMAT AFMT_S16_LE

Definition at line 235 of file Config.h.

Referenced by clSoundUI::SoundInThread(), and clSoundUI::SoundOutThread().

8.34.1.148 #define SUI_SND_BITS 16

Definition at line 236 of file Config.h.

Referenced by clSoundUI::SoundOutThread().

8.34.1.149 #define SUI_SND_DATATYPE signed short

Definition at line 237 of file Config.h.

Referenced by clSoundUI::SoundInThread(), and clSoundUI::SoundOutThread().

8.34.1.150 #define SUI_SND_QUEUESIZE 4096

Definition at line 238 of file Config.h.

Referenced by clSoundUI::SoundOutThread().

8.34.1.151 #define SUI_SAMPLECOUNT 2048

Definition at line 239 of file Config.h.

Referenced by clSoundUI::GetCfg().

8.34.1.152 #define SUI_DEF_DEVICE "/dev/dsp"

Definition at line 240 of file Config.h.

Referenced by clSoundUI::GetCfg().

8.34.1.153 #define SUI_DEF_CHANNELS 2

Definition at line 241 of file Config.h.

Referenced by clSoundUI::GetCfg().

8.34.1.154 #define SUI_DEF_SAMPLERATE 44100

Definition at line 242 of file Config.h.

Referenced by clSoundUI::GetCfg().

8.34.1.155 #define SUI_CFGFILE "soundui.cfg"

Definition at line 243 of file Config.h.

Referenced by clSoundUI::GetCfg().

8.34.1.156 #define SUI_HOSTFILE "soundui.hosts"

Definition at line 244 of file Config.h.

Referenced by clSoundChGUI::clSoundChGUI().

8.34.1.157 #define SUI_FIRST_TIMEOUT 1000

Definition at line 245 of file Config.h.

Referenced by clSoundUI::SoundInThread().

8.34.1.158 #define SUI_IN_TIMEOUT 250

Definition at line 246 of file Config.h.

Referenced by clSoundUI::SoundInThread().

8.34.1.159 #define SUI_VU_TIMEOUT 200

Definition at line 247 of file Config.h.

Referenced by clSoundUI::GetCfg().

8.34.1.160 #define SP_VERSMAJ GLOBAL_VERSMAJ

Definition at line 250 of file Config.h.

Referenced by main().

8.34.1.161 #define SP_VERSMIN GLOBAL_VERSMIN

Definition at line 251 of file Config.h.

Referenced by main().

8.34.1.162 #define SP_VERSPL GLOBAL_VERSPL

Definition at line 252 of file Config.h.

Referenced by main().

8.34.1.163 #define SP_CFGFILE "soundproxy.cfg"

Definition at line 253 of file Config.h.

Referenced by clSoundProxy::clSoundProxy().

8.34.1.164 #define SP_DEF_LOGFILE "log/soundproxy.cfg"

Definition at line 254 of file Config.h.

Referenced by clSoundProxy::clSoundProxy().

8.34.1.165 #define SP_SERV_MAXLEN 255

Definition at line 255 of file Config.h.

8.34.1.166 #define SP_MAXCLIENTS 255

Definition at line 256 of file Config.h.

Referenced by clSoundProxy::clSoundProxy(), and clSoundProxy::FindFreeSlot().

8.34.1.167 #define SP_1ST_MSG_TIMEOUT 1000

Definition at line 257 of file Config.h.

Referenced by clSoundProxy::ServeClientThread(), and clSoundProxy::SoundInThread().

8.34.1.168 #define SP_MSG_TIMEOUT 250

Definition at line 258 of file Config.h.

Referenced by clSoundProxy::ServeClientThread(), and clSoundProxy::SoundInThread().

8.34.1.169 #define SP_WAIT_CONN_TIMEOUT 250

Definition at line 259 of file Config.h.

Referenced by clSoundProxy::WaitConnectThread().

8.34.1.170 #define SP_SCHED_PRIORITY 1

Definition at line 260 of file Config.h.

Referenced by clSoundProxy::ServeClientThread(), and clSoundProxy::SoundInThread().

8.34.1.171 #define SP_BUFFER_SIZE 4096

Definition at line 261 of file Config.h.

Referenced by clSoundProxy::clSoundProxy(), clSoundProxy::ServeClientThread(), and clSoundProxy::SoundInThread().

8.34.1.172 #define BA_CFGFILE "beamaudio.cfg"

Definition at line 264 of file Config.h.

Referenced by clBeamAudio::GetCfg().

8.34.1.173 #define BA_1STREQ_TIMEOUT 5000

Definition at line 265 of file Config.h.

Referenced by clBeamAudio::GetRq().

8.34.1.174 #define BA_RAW1ST_TIMEOUT 1000

Definition at line 266 of file Config.h.

Referenced by clBeamAudio::ConnectStream().

8.34.1.175 #define BA_TIMEOUT 250

Definition at line 267 of file Config.h.

8.34.1.176 #define BA_DEF_FRAGMENT_SIZE 4096

Definition at line 268 of file Config.h.

Referenced by clBeamAudio::GetCfg().

8.34.1.177 #define BA_SCHED_PRIORITY 2

Definition at line 269 of file Config.h.

Referenced by main().

8.34.1.178 #define BAUI_CFGFILE "beamaudioui.cfg"

Definition at line 272 of file Config.h.

Referenced by clBeamAudioUI::clBeamAudioUI().

8.34.1.179 #define BAUI_HOSTFILE "beamaudioui.hosts"

Definition at line 273 of file Config.h.

Referenced by clBeamAudioUI::BuildTable1().

8.34.1.180 #define BAUI_REQ_PROC "beamaudio"

Definition at line 274 of file Config.h.

Referenced by clBeamAudioUI::InitConnection().

8.34.1.181 #define BAUI_1ST_TIMEOUT 5000

Definition at line 275 of file Config.h.

Referenced by clBeamAudioUI::SendSettings().

8.34.1.182 #define BAUI_DEF_SOUNDSPEED "1430.0"

Definition at line 276 of file Config.h.

Referenced by clBeamAudioUI::BuildTable2().

8.34.1.183 #define BAUI_DEF_DIR_RANGE 180.0

Definition at line 277 of file Config.h.

Referenced by clBeamAudioUI::Build().

8.34.1.184 #define BAUI_INT_DATATYPE int

Definition at line 278 of file Config.h.

Referenced by clBeamAudioUI::AllocateBuffers(), clBeamAudioUI::ConvertFromDither(), and clBeamAudioUI::Dither().

8.34.1.185 #define BAUI_SND_FORMAT AFMT_S16_LE

Definition at line 279 of file Config.h.

Referenced by clBeamAudioUI::InitAudio().

8.34.1.186 #define BAUI_SND_BITS 16

Definition at line 280 of file Config.h.

Referenced by clBeamAudioUI::InitAudio().

8.34.1.187 #define BAUI_SND_DATATYPE signed short

Definition at line 281 of file Config.h.

Referenced by clBeamAudioUI::AllocateBuffers(), clBeamAudioUI::AudioOutThread(), clBeamAudioUI::ConvertFromDither(), and clBeamAudioUI::InitAudio().

8.34.1.188 #define LEVEL_CFGFILE "level.cfg"

Definition at line 284 of file Config.h.

Referenced by clLevel::GetCfg().

8.34.1.189 #define LEVEL_1STREQ_TIMEOUT 5000

Definition at line 285 of file Config.h.

Referenced by clLevel::GetRequest().

8.34.1.190 #define LEVEL_RAW1ST_TIMEOUT 1000

Definition at line 286 of file Config.h.

Referenced by clLevel::ConnectStream().

8.34.1.191 #define LEVEL_TIMEOUT 250

Definition at line 287 of file Config.h.

Referenced by clLevel::ReceiveData().

8.34.1.192 #define LEVEL_DEF_FILTER_SIZE 4096

Definition at line 288 of file Config.h.

Referenced by clLevel::GetCfg().

8.34.1.193 #define GUILEV_CFGFILE "guilevel.cfg"

Definition at line 291 of file Config.h.

Referenced by clGUILevel::GetCfg().

8.34.1.194 #define GUILEV_HOSTFILE "guilevel.hosts"

Definition at line 292 of file Config.h.

Referenced by clGUILevel::BuildTable1().

8.34.1.195 #define GUILEV_REQ_PROC "level"

Definition at line 293 of file Config.h.

Referenced by clGUILevel::ConnectToServer().

8.34.1.196 #define LOCATE_CFGFILE "locate.cfg"

Definition at line 296 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.197 #define LOCATE_LOGFILE "log/locate.log"

Definition at line 297 of file Config.h.

Referenced by clLocate::clLocate().

8.34.1.198 #define LOCATE_SENSOR_LIST "locate.sensors"

Definition at line 298 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.199 #define LOCATE_DIR_PROC "direction3"

Definition at line 299 of file Config.h.

Referenced by clSubLocate::ConnectDir().

8.34.1.200 #define LOCATE_TIMEOUT 250

Definition at line 300 of file Config.h.

Referenced by clLocate::Main().

8.34.1.201 #define LOCATE_DEF_WINDOWSIZE 4096

Definition at line 301 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.202 #define LOCATE_DEF_SOUNDSPEED 1430.0

Definition at line 302 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.203 #define LOCATE_DEF_LOWFREQ 10.0

Definition at line 303 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.204 #define LOCATE_DEF_INTTIME 1.0

Definition at line 304 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.205 #define LOCATE_DEF_SCALING 0

Definition at line 305 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.206 #define LOCATE_DEF_SCALINGEXP 2.0

Definition at line 306 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.207 #define LOCATE_DEF_REMOVENOISE 0

Definition at line 307 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.208 #define LOCATE_DEF_ALPHA 2.0

Definition at line 308 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.209 #define LOCATE_DEF_MEANLENGTH 10

Definition at line 309 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.210 #define LOCATE_DEF_GAPLENGTH 3

Definition at line 310 of file Config.h.

Referenced by clLocate::Initialize().

8.34.1.211 #define GUILOC_CFGFILE "guilocate.cfg"

Definition at line 313 of file Config.h.

Referenced by clGUILocate::GetCfg().

8.34.1.212 #define GUILOC_HOSTFILE "guilocate.hosts"

Definition at line 314 of file Config.h.

Referenced by clGUILocate::BuildTable1().

8.34.1.213 #define GUITRANS_CFGFILE "guitrans.cfg"

Definition at line 317 of file Config.h.

Referenced by clGUITransient::GetCfg().

8.34.1.214 #define GUITRANS_HOSTFILE "guitrans.hosts"

Definition at line 318 of file Config.h.

Referenced by clGUITransient::BuildTable1().

8.35 ConvPic.cc File Reference

`#include <cstdio>`

`#include <cstring>`

`#include <climits>`

`#include <ctime>`

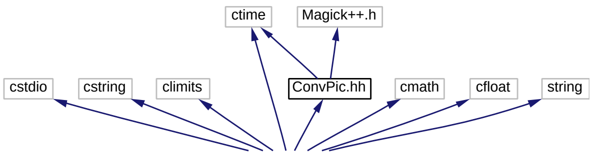
`#include <cmath>`

`#include <cfloat>`

`#include <string>`

`#include "ConvPic.hh"`

Include dependency graph for ConvPic.cc:



Functions

- int **main** (int argc, char ∗argv[ ])

Variables

- **clConvPic ConvPic**

8.35.1 Function Documentation

8.35.1.1 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 37 of file ConvPic.cc.

References ConvPic, and clConvPic::Main().

8.35.2 Variable Documentation

8.35.2.1 clConvPic ConvPic

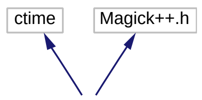
Definition at line 34 of file ConvPic.cc.

Referenced by main().

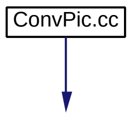
8.36 ConvPic.hh File Reference

```
#include <ctime>
#include <Magick++.h>
```

Include dependency graph for ConvPic.hh:



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



Namespaces

- namespace **std**
- namespace **Magick**

Compounds

- struct **_stLOFARInfo**

  Information stored in saved LOFAR .tif.inf.

- struct **_stDEMONInfo**

  Information stored in saved DEMON .tif.inf.

- struct **_stSGramInfo**

  Information stored in saved spectrogram .tif.inf.

- struct **_stTBearInfo**

  Information stored in saved bearing-time .tif.inf.

- union **_uPicInfo**

  Combination of information in all .inf files.

- class **clConvPic**

  Conversion of .tif + .tif.inf to image containing information in the .inf file.

Defines

- #define **CP_FONT_NAME** "helvetica"
- #define **CP_FONT_SMALL** 10
- #define **CP_FONT_NORMAL** 12
- #define **CP_FONT_LARGE** 16
- #define **CP_PAGE_SPEC** "+100+100"
- #define **CP_MARGIN_L** 100
- #define **CP_MARGIN_T** 100
- #define **CP_MARGIN_R** 50
- #define **CP_MARGIN_B** 50
- #define **CP_TICKL_LEN** 10
- #define **CP_TICKS_LEN** 5

Typedefs

- typedef **_stLOFARInfo stLOFARInfo**

  Information stored in saved LOFAR .tif.inf.

- typedef **_stLOFARInfo** * **stpLOFARInfo**

  Information stored in saved LOFAR .tif.inf.

- typedef **_stDEMONInfo stDEMONInfo**

  Information stored in saved DEMON .tif.inf.

- typedef **_stDEMONInfo** * **stpDEMONInfo**

  Information stored in saved DEMON .tif.inf.

- typedef **_stSGramInfo stSGramInfo**

  Information stored in saved spectrogram .tif.inf.

- typedef **_stSGramInfo** * **stpSGramInfo**

  Information stored in saved spectrogram .tif.inf.

- typedef **_stTBearInfo stTBearInfo**

  Information stored in saved bearing-time .tif.inf.

- typedef **_stTBearInfo** * **stpTBearInfo**

  Information stored in saved bearing-time .tif.inf.

- typedef **_uPicInfo utPicInfo**

  Combination of information in all .inf files.

- typedef **_uPicInfo** * **utpPicInfo**

  Combination of information in all .inf files.

Enumerations

- enum { **CP_TYPE_LOFAR**, **CP_TYPE_DEMON**, **CP_TYPE_SGRAM**, **CP_TYPE_TBEAR** }

  Types for different kinds of saved images.

8.36.1 Define Documentation

8.36.1.1 #define CP_FONT_NAME "helvetica"

Definition at line 30 of file ConvPic.hh.

Referenced by clConvPic::LoadAndCreate().

8.36.1.2 #define CP_FONT_SMALL 10

Definition at line 31 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), clConvPic::DrawLOFARInfo(), clConvPic::DrawSGramInfo(), and clConvPic::DrawTBearInfo().

8.36.1.3 #define CP_FONT_NORMAL 12

Definition at line 32 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), clConvPic::DrawLOFARInfo(), clConvPic::DrawSGramInfo(), and clConvPic::DrawTBearInfo().

8.36.1.4 #define CP_FONT_LARGE 16

Definition at line 33 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), clConvPic::DrawLOFARInfo(), clConvPic::DrawSGramInfo(), and clConvPic::DrawTBearInfo().

8.36.1.5 #define CP_PAGE_SPEC "+100+100"

Definition at line 34 of file ConvPic.hh.

Referenced by clConvPic::LoadAndCreate().

8.36.1.6 #define CP_MARGIN_L 100

Definition at line 35 of file ConvPic.hh.

Referenced by clConvPic::CopySource(), clConvPic::DrawDEMONInfo(), clConvPic::DrawLOFARInfo(), clConvPic::DrawSGramInfo(), clConvPic::DrawTBearInfo(), and clConvPic::LoadAndCreate().

8.36.1.7 #define CP_MARGIN_T 100

Definition at line 36 of file ConvPic.hh.

Referenced by clConvPic::CopySource(), clConvPic::DrawDEMONInfo(), clConvPic::DrawLOFARInfo(), clConvPic::DrawSGramInfo(), clConvPic::DrawTBearInfo(), and clConvPic::LoadAndCreate().

8.36.1.8 #define CP_MARGIN_R 50

Definition at line 37 of file ConvPic.hh.

Referenced by clConvPic::LoadAndCreate().

8.36.1.9 #define CP_MARGIN_B 50

Definition at line 38 of file ConvPic.hh.

Referenced by clConvPic::LoadAndCreate().

8.36.1.10 #define CP_TICKL_LEN 10

Definition at line 39 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), clConvPic::DrawLOFARInfo(), clConvPic::DrawSGramInfo(), and clConvPic::DrawTBearInfo().

8.36.1.11 #define CP_TICKS_LEN 5

Definition at line 40 of file ConvPic.hh.

Referenced by clConvPic::DrawDEMONInfo(), clConvPic::DrawLOFARInfo(), clConvPic::DrawSGramInfo(), and clConvPic::DrawTBearInfo().

8.36.2 Typedef Documentation

8.36.2.1 typedef struct _stLOFARInfo stLOFARInfo

Information stored in saved LOFAR .tif.inf.

8.36.2.2 typedef struct _stLOFARInfo * stpLOFARInfo

Information stored in saved LOFAR .tif.inf.

8.36.2.3 typedef struct _stDEMONInfo stDEMONInfo

Information stored in saved DEMON .tif.inf.

8.36.2.4 typedef struct _stDEMONInfo * stpDEMONInfo

Information stored in saved DEMON .tif.inf.

8.36.2.5 typedef struct _stSGramInfo stSGramInfo

Information stored in saved spectrogram .tif.inf.

8.36.2.6 typedef struct _stSGramInfo * stpSGramInfo

Information stored in saved spectrogram .tif.inf.

8.36.2.7 typedef struct _stTBearInfo stTBearInfo

Information stored in saved bearing-time .tif.inf.

8.36.2.8 typedef struct _stTBearInfo * stpTBearInfo

Information stored in saved bearing-time .tif.inf.

8.36.2.9 typedef union _uPicInfo utPicInfo

Combination of information in all .inf files.

8.36.2.10 typedef union _uPicInfo * utpPicInfo

Combination of information in all .inf files.

8.36.3 Enumeration Type Documentation

8.36.3.1 anonymous enum

Types for different kinds of saved images.

Enumeration values:

- **CP_TYPE_LOFAR**
- **CP_TYPE_DEMON**
- **CP_TYPE_SGRAM**
- **CP_TYPE_TBEAR**

Definition at line 50 of file ConvPic.hh.

8.37 CorrDipole.cc File Reference

`#include <stdio.h>`

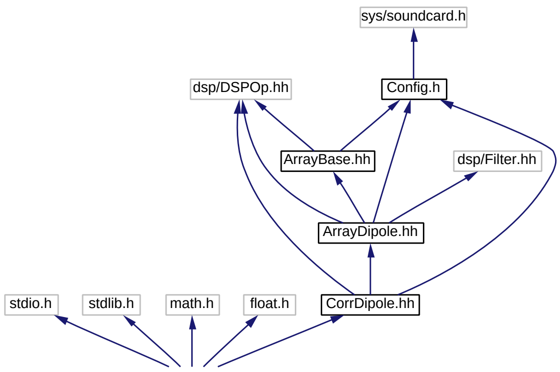
`#include <stdlib.h>`

`#include <math.h>`

`#include <float.h>`

`#include "CorrDipole.hh"`

Include dependency graph for CorrDipole.cc:



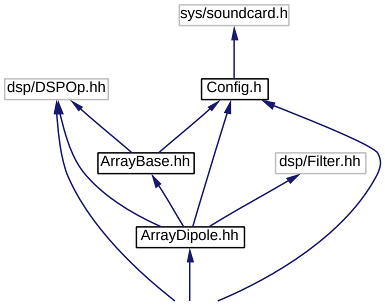
8.38 CorrDipole.hh File Reference

`#include <dsp/DSPOp.hh>`

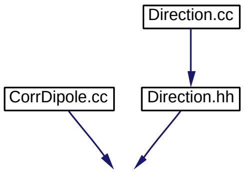
`#include "Config.h"`

`#include "ArrayDipole.hh"`

Include dependency graph for CorrDipole.hh:



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



Compounds

- class **clCorrDipole**

  Correlator class for dipole array.

8.39 Direction.cc File Reference

`#include <pthread.h>`

`#include <stdio.h>`

`#include <stdlib.h>`

`#include <unistd.h>`

`#include <limits.h>`

`#include <string.h>`

`#include <signal.h>`

`#include <errno.h>`

`#include <math.h>`

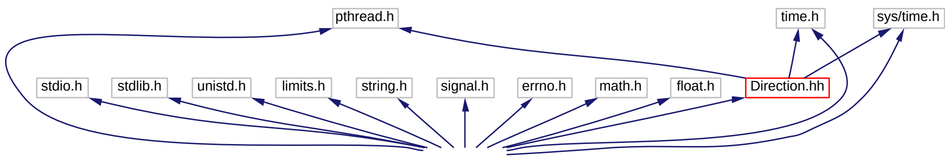
`#include <float.h>`

`#include <time.h>`

`#include <sys/time.h>`

`#include "Direction.hh"`

Include dependency graph for Direction.cc:



Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char *argv[])
- void * **ThrReceiveData** (void *vpParam)
- void * **ThrProcessData** (void *vpParam)
- void * **ThrSendResults** (void *vpParam)

Variables

- bool **bDebug** = false
- bool **bDaemon** = false
- const char * **cpaArrayTypes** []
- **clDirection** * **Dir**

8.39.1 Function Documentation

8.39.1.1 void SigHandler (int *iSigNum*)

Definition at line 46 of file Direction.cc.

References bDebug.

8.39.1.2 int main (int *argc*, char * *argv*[ ])

Definition at line 62 of file Direction.cc.

References bDaemon, bDebug, Dir, clDirection::Exec(), and SigHandler().

8.39.1.3 void* ThrReceiveData (void * *vpParam*)

Definition at line 88 of file Direction.cc.

References Dir, and clDirection::ReceiveData().

Referenced by clDirection::Exec().

8.39.1.4 void* ThrProcessData (void * *vpParam*)

Definition at line 94 of file Direction.cc.

References Dir, and clDirection::ProcessData().

Referenced by clDirection::Exec().

8.39.1.5 void* ThrSendResults (void * *vpParam*)

Definition at line 100 of file Direction.cc.

References Dir, and clDirection::SendResults().

Referenced by clDirection::Exec().

8.39.2 Variable Documentation

8.39.2.1 bool bDebug = false [static]

Definition at line 39 of file Direction.cc.

Referenced by clDirection::Exec(), clDirection::GetFirstMsg(), clDirection::GetRequest(), clDirection::InitArray(), main(), clDirection::ProcessData(), clDirection::ReceiveData(), clDirection::SendResults(), SigHandler(), and clDirection::StopTiming().

8.39.2.2 bool bDaemon = false [static]

Definition at line 40 of file Direction.cc.

Referenced by clDirection::Exec(), clDirection::InitArray(), and main().

8.39.2.3 const char* cpaArrayTypes[] [static]

Initial value:

```
{ "Dipole", "Triangle", "Line",
   "Plane", "Cylinder", "Sphere" }
```

Definition at line 41 of file Direction.cc.

Referenced by clDirection::InitArray().

8.39.2.4 clDirection* Dir [static]

Definition at line 43 of file Direction.cc.

Referenced by main(), ThrProcessData(), ThrReceiveData(), and ThrSend-Results().

8.40 Direction.hh File Reference

`#include <pthread.h>`

`#include <time.h>`

`#include <sys/time.h>`

`#include <Alloc.hh>`

`#include <Condition.hh>`

`#include <Mutex.hh>`

`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

`#include "LocalMsg.h"`

`#include "ArrayBase.hh"`

`#include "ArrayDipole.hh"`

`#include "BeamDipole.hh"`

`#include "CorrDipole.hh"`

`#include "CfgFile.hh"`

`#include "Messages.hh"`

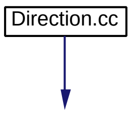
`#include "RemoveNoise.hh"`

`#include "SockClie.hh"`

`#include "SockOp.hh"`

Include dependency graph for Direction.hh:

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



Compounds

- class **clDirection**
  Direction calculation server.

Defines

- #define **AB_COMPILE**

Enumerations

- enum {
  DIR_ARRAY_TYPE_DIPOLE = 0, **DIR_ARRAY_TYPE_-TRIANGLE** = 1, **DIR_ARRAY_TYPE_LINE** = 2, **DIR_-ARRAY_TYPE_PLANE** = 3,
  DIR_ARRAY_TYPE_CYLINDER = 4, **DIR_ARRAY_TYPE_-SPHERE** = 5 }

  Array types.

8.40.1 Define Documentation

8.40.1.1 #define AB_COMPILE

Definition at line 34 of file Direction.hh.

8.40.2 Enumeration Type Documentation

8.40.2.1 anonymous enum

Array types.

Enumeration values:

DIR_ARRAY_TYPE_DIPOLE

DIR_ARRAY_TYPE_TRIANGLE

DIR_ARRAY_TYPE_LINE

DIR_ARRAY_TYPE_PLANE

DIR_ARRAY_TYPE_CYLINDER

DIR_ARRAY_TYPE_SPHERE

Definition at line 53 of file Direction.hh.

8.41 Direction2.cc File Reference

`#include <stdio.h>`

`#include <signal.h>`

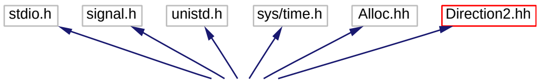
`#include <unistd.h>`

`#include <sys/time.h>`

`#include <Alloc.hh>`

`#include "Direction2.hh"`

Include dependency graph for Direction2.cc:



Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char *argv[ ])

Variables

- bool **bDebug**
- bool **bDaemon**
- **clDirection2** * **Direction2**

8.41.1 Function Documentation

8.41.1.1 void SigHandler (int *iSigNum*)

Definition at line 38 of file Direction2.cc.

References bDebug.

8.41.1.2 int main (int *argc*, char * *argv*[ ])

Definition at line 54 of file Direction2.cc.

References bDaemon, bDebug, Direction2, clDirection2::Main(), and SigHandler().

8.41.2 Variable Documentation

8.41.2.1 bool bDebug [static]

Definition at line 33 of file Direction2.cc.

Referenced by clDirection2::InitDir(), clDirection2::Main(), main(), clDirection2::ProcessLoop(), and SigHandler().

8.41.2.2 bool bDaemon [static]

Definition at line 34 of file Direction2.cc.

Referenced by main().

8.41.2.3 clDirection2* Direction2

Definition at line 35 of file Direction2.cc.

Referenced by main().

8.42 Direction2.hh File Reference

`#include <limits.h>`

`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

`#include "LocalMsg.h"`

`#include "CfgFile.hh"`

`#include "Messages.hh"`

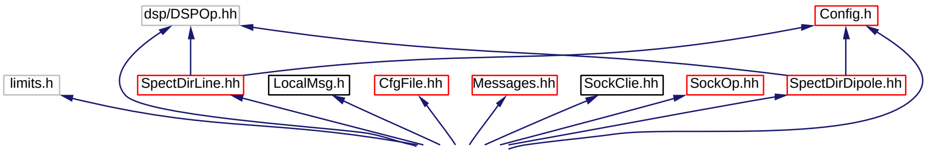
`#include "SockClie.hh"`

`#include "SockOp.hh"`

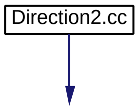
`#include "SpectDirDipole.hh"`

`#include "SpectDirLine.hh"`

Include dependency graph for Direction2.hh:



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



Compounds

- class **clDirection2**

  Spectrum based direction server.

Enumerations

- enum {

  DIR2_ARRAY_TYPE_DIPOLE = 0, **DIR2_ARRAY_TYPE_-TRIANGLE** = 1, **DIR2_ARRAY_TYPE_LINE** = 2, **DIR2_-ARRAY_TYPE_PLANE** = 3,

DIR2_ARRAY_TYPE_CYLINDER = 4, **DIR2_ARRAY_TYPE_-SPHERE** = 5 }

Array types.

8.42.1 Enumeration Type Documentation

8.42.1.1 anonymous enum

Array types.

Enumeration values:

DIR2_ARRAY_TYPE_DIPOLE

DIR2_ARRAY_TYPE_TRIANGLE

DIR2_ARRAY_TYPE_LINE

DIR2_ARRAY_TYPE_PLANE

DIR2_ARRAY_TYPE_CYLINDER

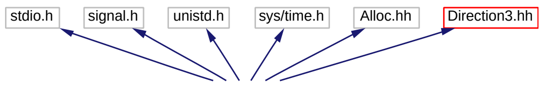
DIR2_ARRAY_TYPE_SPHERE

Definition at line 44 of file Direction2.hh.

8.43 Direction3.cc File Reference

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <sys/time.h>
#include <Alloc.hh>
#include "Direction3.hh"
```

Include dependency graph for Direction3.cc:



Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char ∗argv[ ])

Variables

- bool **bDebug**
- bool **bDaemon**
- **clDirection3** ∗ **Direction3**

8.43.1 Function Documentation

8.43.1.1 void SigHandler (int *iSigNum*)

Definition at line 38 of file Direction3.cc.

References bDebug.

8.43.1.2 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 54 of file Direction3.cc.

References bDaemon, bDebug, Direction3, clDirection3::Main(), and SigHandler().

8.43.2 Variable Documentation

8.43.2.1 bool bDebug [static]

Definition at line 33 of file Direction3.cc.

Referenced by clDirection3::InitDir(), clDirection3::Main(), main(), clDirection3::ProcessLoop(), and SigHandler().

8.43.2.2 bool bDaemon [static]

Definition at line 34 of file Direction3.cc.

Referenced by main().

8.43.2.3 clDirection3* Direction3

Definition at line 35 of file Direction3.cc.

Referenced by main().

8.44 Direction3.hh File Reference

`#include <limits.h>`

`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

`#include "LocalMsg.h"`

`#include "CfgFile.hh"`

`#include "Messages.hh"`

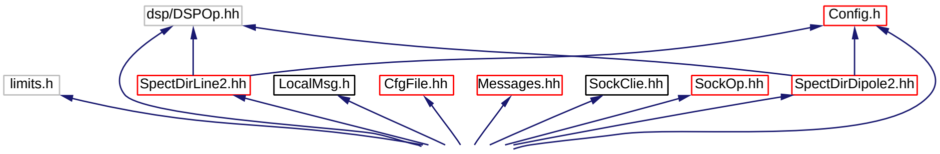
`#include "SockClie.hh"`

`#include "SockOp.hh"`

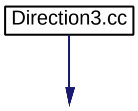
`#include "SpectDirDipole2.hh"`

`#include "SpectDirLine2.hh"`

Include dependency graph for Direction3.hh:



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



Compounds

- class **clDirection3**

  Spectrum based direction server.

Enumerations

- enum {

  DIR3_ARRAY_TYPE_DIPOLE = 0, **DIR3_ARRAY_TYPE_TRIANGLE** = 1, **DIR3_ARRAY_TYPE_LINE** = 2, **DIR3_ARRAY_TYPE_PLANE** = 3,

DIR3_ARRAY_TYPE_CYLINDER = 4, **DIR3_ARRAY_TYPE_-SPHERE** = 5 }

Array types.

8.44.1 Enumeration Type Documentation

8.44.1.1 anonymous enum

Array types.

Enumeration values:

DIR3_ARRAY_TYPE_DIPOLE

DIR3_ARRAY_TYPE_TRIANGLE

DIR3_ARRAY_TYPE_LINE

DIR3_ARRAY_TYPE_PLANE

DIR3_ARRAY_TYPE_CYLINDER

DIR3_ARRAY_TYPE_SPHERE

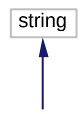
Definition at line 44 of file Direction3.hh.

8.45 Extensions.cc File Reference

8.46 Extensions.hh File Reference

`#include <string>`

Include dependency graph for Extensions.hh:



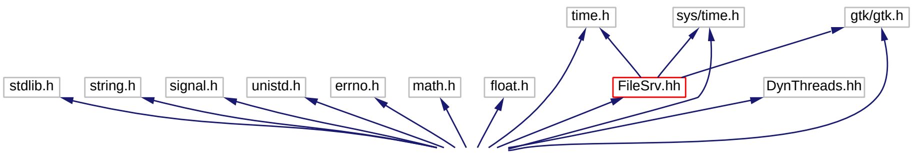
Compounds

- class **estring**

8.47 FileSrv.cc File Reference

```
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <unistd.h>
#include <errno.h>
#include <math.h>
#include <float.h>
#include <time.h>
#include <sys/time.h>
#include <gtk/gtk.h>
#include <DynThreads.hh>
#include "FileSrv.hh"
```

Include dependency graph for FileSrv.cc:



Functions

- clDynThreads< **clFileSrv** > **FileSrvThreads** (**FileSrv**)
- int **main** (int argc, char *argv[])
- gboolean **WrapOnDelete** (GtkWidget *gwSender, GdkEvent *geEvent, gpointer gpData)
- void **WrapOnFileSelectOkClick** (GtkButton *gbSender, gpointer gpData)
- void **WrapOnFileSelectCancelClick** (GtkButton *gbSender, gpointer gpData)
- void **WrapOnBrowseClick** (GtkButton *gbSender, gpointer gpData)
- void **WrapOnPlayStopToggle** (GtkToggleButton *gtbSender, gpointer gpData)
- void **WrapOnPositionChange** (GtkAdjustment *gaSender, gpointer gpData)

Variables

- const char * **cpWindowTxt** = "File server"

- const char ∗ **cpFileSelectTxt** = "Select file"
- const char ∗ **cpFileTxt** = "Filename"
- const char ∗ **cpBrowseTxt** = "Browse..."
- const char ∗ **cpPlayStopTxt** [ ] = { "Play", "Stop" }
- const char ∗ **cpPositionTxt** = "Position"
- **clFileSrv FileSrv**

8.47.1 Function Documentation

8.47.1.1 clDynThreads<clFileSrv> FileSrvThreads (FileSrv)

Referenced by clFileSrv::Main(), clFileSrv::ReaderThread(), clFileSrv::ServeClientThread(), and clFileSrv::ServerThread().

8.47.1.2 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 54 of file FileSrv.cc.

References FileSrv, and clFileSrv::Main().

8.47.1.3 gboolean WrapOnDelete (GtkWidget ∗ *gwSender*, GdkEvent ∗ *geEvent*, gpointer *gpData*)

Definition at line 62 of file FileSrv.cc.

References FileSrv, and clFileSrv::OnDelete().

8.47.1.4 void WrapOnFileSelectOkClick (GtkButton ∗ *gbSender*, gpointer *gpData*)

Definition at line 68 of file FileSrv.cc.

References FileSrv, and clFileSrv::OnFileSelectOkClick().

Referenced by clFileSrv::ConnectSignals().

8.47.1.5 void WrapOnFileSelectCancelClick (GtkButton ∗ *gbSender*, gpointer *gpData*)

Definition at line 74 of file FileSrv.cc.

References FileSrv, and clFileSrv::OnFileSelectCancelClick().

Referenced by clFileSrv::ConnectSignals().

8.47.1.6 void WrapOnBrowseClick (GtkButton ∗ *gbSender*, gpointer *gpData*)

Definition at line 80 of file FileSrv.cc.

References FileSrv, and clFileSrv::OnBrowseClick().

Referenced by clFileSrv::ConnectSignals().

8.47.1.7 void WrapOnPlayStopToggle (GtkToggleButton * *gtbSender*, gpointer *gpData*)

Definition at line 86 of file FileSrv.cc.

References FileSrv, and clFileSrv::OnPlayStopToggle().

Referenced by clFileSrv::ConnectSignals().

8.47.1.8 void WrapOnPositionChange (GtkAdjustment * *gaSender*, gpointer *gpData*)

Definition at line 92 of file FileSrv.cc.

References FileSrv, and clFileSrv::OnPositionChange().

8.47.2 Variable Documentation

8.47.2.1 const char* cpWindowTxt = "File server" `[static]`

Definition at line 40 of file FileSrv.cc.

Referenced by clFileSrv::Build(), and clFileSrv::Main().

8.47.2.2 const char* cpFileSelectTxt = "Select file" `[static]`

Definition at line 41 of file FileSrv.cc.

Referenced by clFileSrv::Build().

8.47.2.3 const char* cpFileTxt = "Filename" `[static]`

Definition at line 43 of file FileSrv.cc.

Referenced by clFileSrv::BuildTable1().

8.47.2.4 const char* cpBrowseTxt = "Browse..." `[static]`

Definition at line 44 of file FileSrv.cc.

Referenced by clFileSrv::BuildTable1().

8.47.2.5 const char* cpPlayStopTxt[] = { "Play", "Stop" } [static]

Definition at line 46 of file FileSrv.cc.

Referenced by clFileSrv::BuildTable2().

8.47.2.6 const char* cpPositionTxt = "Position" [static]

Definition at line 47 of file FileSrv.cc.

Referenced by clFileSrv::BuildTable2().

8.47.2.7 clFileSrv FileSrv

Definition at line 50 of file FileSrv.cc.

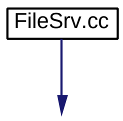
Referenced by main(), WrapOnBrowseClick(), WrapOnDelete(), WrapOnFileSelectCancelClick(), WrapOnFileSelectOkClick(), WrapOnPlayStopToggle(), and WrapOnPositionChange().

8.48 FileSrv.hh File Reference

```
#include <time.h>
#include <sys/time.h>
#include <gtk/gtk.h>
#include <sndfile.h>
#include <Alloc.hh>
#include <Mutex.hh>
#include <Condition.hh>
#include <dsp/ReBufferT.hh>
#include "Config.h"
#include "CfgFile.hh"
#include "GtkUtils.hh"
#include "Messages.hh"
#include "SockServ.hh"
#include "SockOp.hh"
```

Include dependency graph for FileSrv.hh:

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



Compounds

- class **clFileSrv**

  Input server for playback of previously recorded data.

Defines

- #define **FS_VER_MAJ** GLOBAL_VERSMAJ
- #define **FS_VER_MIN** GLOBAL_VERSMIN
- #define **FS_VER_PL** GLOBAL_VERSPL
- #define **FS_WSPACING** 8
- #define **FS_ACCEPT_TIMEOUT** 250

8.48.1 Define Documentation

8.48.1.1 #define FS_VER_MAJ GLOBAL_VERSMAJ

Definition at line 47 of file FileSrv.hh.

Referenced by clFileSrv::Main().

8.48.1.2 #define FS_VER_MIN GLOBAL_VERSMIN

Definition at line 48 of file FileSrv.hh.

Referenced by clFileSrv::Main().

8.48.1.3 #define FS_VER_PL GLOBAL_VERSPL

Definition at line 49 of file FileSrv.hh.

Referenced by clFileSrv::Main().

8.48.1.4 #define FS_WSPACING 8

Definition at line 50 of file FileSrv.hh.

Referenced by clFileSrv::Build(), clFileSrv::BuildTable1(), and clFileSrv::BuildTable2().

8.48.1.5 #define FS_ACCEPT_TIMEOUT 250

Definition at line 51 of file FileSrv.hh.

Referenced by clFileSrv::ServerThread().

8.49 FrameBuf.cc File Reference

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <float.h>

#include <time.h>

#include <unistd.h>

#include <tiffio.h>

#include <gtk/gtk.h>

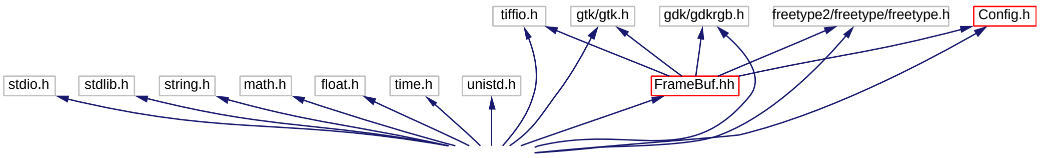
#include <gdk/gdkrgb.h>

#include <freetype2/freetype/freetype.h>

#include "Config.h"

#include "FrameBuf.hh"

Include dependency graph for FrameBuf.cc:



8.50 FrameBuf.hh File Reference

`#include <gtk/gtk.h>`

`#include <gdk/gdkrgb.h>`

`#include <tiffio.h>`

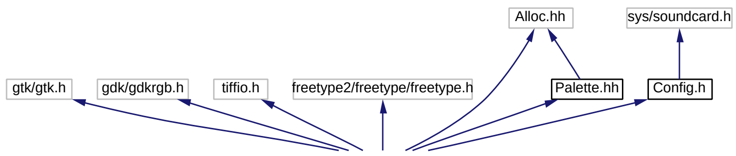
`#include <freetype2/freetype/freetype.h>`

`#include <Alloc.hh>`

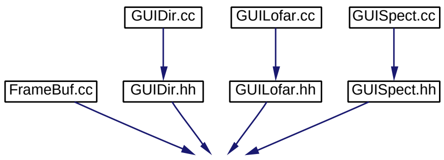
`#include "Config.h"`

`#include "Palette.hh"`

Include dependency graph for FrameBuf.hh:



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



Compounds

- class **clFrameBuf**

  Framebuffer class.

Defines

- #define **FB_TIFF_MODE** "w"
- #define **FB_TIFF_RESOLUTION** 100
- #define **FB_TIFF_FONTSIZE** 12
- #define **FB_TIFF_DATELEN** 19

Enumerations

- enum { **FB_TYPE_NONE** = 0, **FB_TYPE_LINE** = 1, **FB_TYPE_COLUMN** = 2 }

  Frame buffer types.

- enum {

  FB_TIFF_COMPRESS_NONE = COMPRESSION_NONE, **FB_TIFF_COMPRESS_RLE** = COMPRESSION_CCITTRLE, **FB_TIFF_COMPRESS_LZW** = COMPRESSION_LZW, **FB_TIFF_COMPRESS_JPEG** = COMPRESSION_JPEG,

  FB_TIFF_COMPRESS_DEFLATE = COMPRESSION_ADOBE_DEFLATE }

  TIFF CODECs.

- enum { **FB_TIFF_CONT_NO**, **FB_TIFF_CONT_VERTICAL**, **FB_TIFF_CONT_HORIZONTAL** }

  TIFF save mode.

8.50.1 Define Documentation

8.50.1.1 #define FB_TIFF_MODE "w"

Definition at line 39 of file FrameBuf.hh.

Referenced by clSpectGUI::SaveInfo(), clGUILofar::SaveInfo(), clGUIDir::SaveInfo(), clFrameBuf::SaveToFile(), and clFrameBuf::StartSaveToFile().

8.50.1.2 #define FB_TIFF_RESOLUTION 100

Definition at line 40 of file FrameBuf.hh.

Referenced by clFrameBuf::clFrameBuf(), clFrameBuf::DrawText(), and clFrameBuf::SetTiffTags().

8.50.1.3 #define FB_TIFF_FONTSIZE 12

Definition at line 41 of file FrameBuf.hh.

Referenced by clFrameBuf::clFrameBuf().

8.50.1.4 #define FB_TIFF_DATELEN 19

Definition at line 42 of file FrameBuf.hh.

Referenced by clFrameBuf::SetTiffTags().

8.50.2 Enumeration Type Documentation

8.50.2.1 anonymous enum

Frame buffer types.

Enumeration values:

- **FB_TYPE_NONE**
- **FB_TYPE_LINE**
- **FB_TYPE_COLUMN**

Definition at line 48 of file FrameBuf.hh.

8.50.2.2 anonymous enum

TIFF CODECs.

Enumeration values:

- **FB_TIFF_COMPRESS_NONE**
- **FB_TIFF_COMPRESS_RLE**
- **FB_TIFF_COMPRESS_LZW**
- **FB_TIFF_COMPRESS_JPEG**
- **FB_TIFF_COMPRESS_DEFLATE**

Definition at line 58 of file FrameBuf.hh.

8.50.2.3 anonymous enum

TIFF save mode.

Enumeration values:

- **FB_TIFF_CONT_NO**
- **FB_TIFF_CONT_VERTICAL**
- **FB_TIFF_CONT_HORIZONTAL**

Definition at line 70 of file FrameBuf.hh.

8.51 FreqBeamDipole.cc File Reference

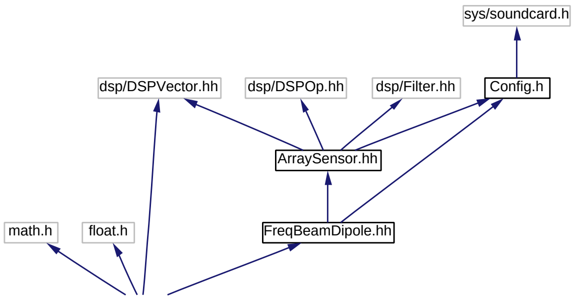
`#include <math.h>`

`#include <float.h>`

`#include <dsp/DSPVector.hh>`

`#include "FreqBeamDipole.hh"`

Include dependency graph for FreqBeamDipole.cc:

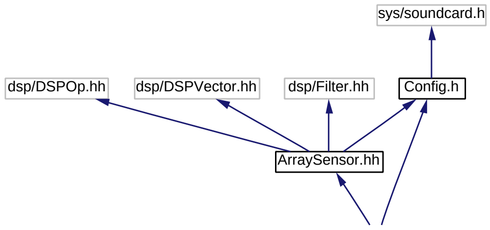


8.52 FreqBeamDipole.hh File Reference

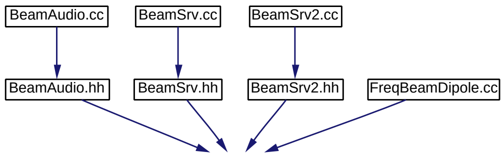
`#include "Config.h"`

`#include "ArraySensor.hh"`

Include dependency graph for FreqBeamDipole.hh:



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



Compounds

- class **clFreqBeamDipole**

  Frequency-domain beamforming for dipole array.

8.53 FreqBeamLine.cc File Reference

`#include <math.h>`

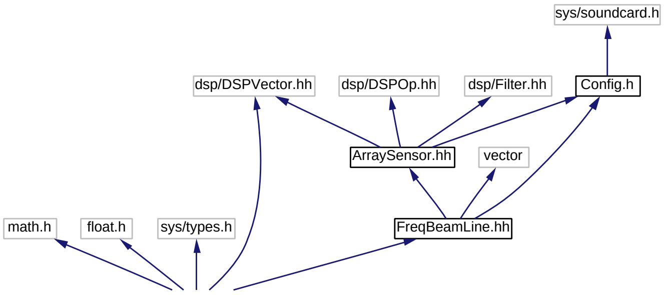
`#include <float.h>`

`#include <sys/types.h>`

`#include <dsp/DSPVector.hh>`

`#include "FreqBeamLine.hh"`

Include dependency graph for FreqBeamLine.cc:



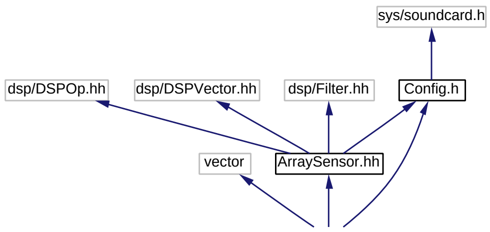
8.54 FreqBeamLine.hh File Reference

`#include <vector>`

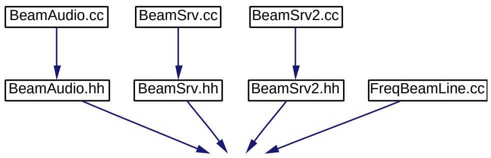
`#include "Config.h"`

`#include "ArraySensor.hh"`

Include dependency graph for FreqBeamLine.hh:



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



Compounds

- class **clFreqBeamLine**

  Frequency-domain beamforming for line array.

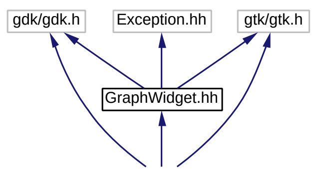
8.55 GraphWidget.cc File Reference

`#include <gdk/gdk.h>`

`#include <gtk/gtk.h>`

`#include "GraphWidget.hh"`

Include dependency graph for GraphWidget.cc:



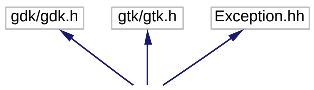
8.56 GraphWidget.hh File Reference

`#include <gdk/gdk.h>`

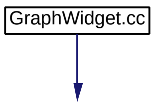
`#include <gtk/gtk.h>`

`#include "Exception.hh"`

Include dependency graph for GraphWidget.hh:



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



Compounds

- class **clGraphWidget**
- class **clGraphWidget.clXGraphWidget**

Defines

- #define **GRAPHWIDGET_MARGIN_TOP** 10
- #define **GRAPHWIDGET_MARGIN_LEFT** 50
- #define **GRAPHWIDGET_MARGIN_BOTTOM** 25
- #define **GRAPHWIDGET_MARGIN_RIGHT** 10

8.56.1 Define Documentation

8.56.1.1 #define GRAPHWIDGET_MARGIN_TOP 10

Definition at line 32 of file GraphWidget.hh.

Referenced by clGraphWidget::clGraphWidget().

8.56.1.2 #define GRAPHWIDGET_MARGIN_LEFT 50

Definition at line 33 of file GraphWidget.hh.

Referenced by clGraphWidget::clGraphWidget().

8.56.1.3 #define GRAPHWIDGET_MARGIN_BOTTOM 25

Definition at line 34 of file GraphWidget.hh.

Referenced by clGraphWidget::clGraphWidget().

8.56.1.4 #define GRAPHWIDGET_MARGIN_RIGHT 10

Definition at line 35 of file GraphWidget.hh.

Referenced by clGraphWidget::clGraphWidget().

8.57 GtkUtils.cc File Reference

`#include <stdlib.h>`

`#include <stdio.h>`

`#include <string.h>`

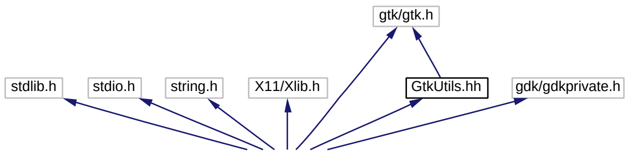
`#include <X11/Xlib.h>`

`#include <gtk/gtk.h>`

`#include <gdk/gdkprivate.h>`

`#include "GtkUtils.hh"`

Include dependency graph for GtkUtils.cc:



Functions

- void **GListFreeItem** (gpointer gpData, gpointer gpUserData)

8.57.1 Function Documentation

8.57.1.1 void GListFreeItem (gpointer *gpData*, gpointer *gpUserData*)

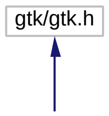
Definition at line 33 of file GtkUtils.cc.

Referenced by clGtkUtils::ComboListFromFile().

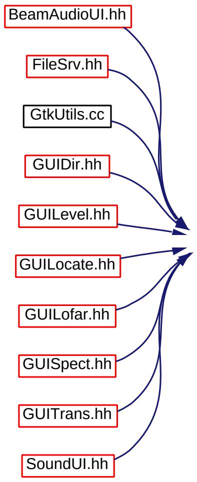
8.58 GtkUtils.hh File Reference

```
#include <gtk/gtk.h>
```

Include dependency graph for GtkUtils.hh:



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



Compounds

- class **clGtkUtils**

  Gtk+ utility class.

Defines

- #define **GU_TXTBUFSIZE** 1024
- #define **EVENT_METHOD**(i, x) GTK_WIDGET_GET_CLASS(GTK_OBJECT(i)) $\rightarrow$ x

8.58.1 Define Documentation

8.58.1.1 #define GU_TXTBUFSIZE 1024

Definition at line 30 of file GtkUtils.hh.

Referenced by clGtkUtils::ComboListFromFile().

8.58.1.2 #define EVENT_METHOD(i, x) GTK_WIDGET_GET_CLASS(GTK_OBJECT(i)) → x

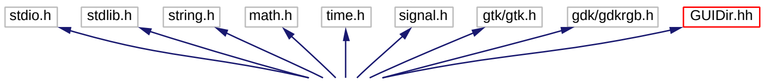
Definition at line 34 of file GtkUtils.hh.

Referenced by clGtkUtils::ConnectMotionEvent().

8.59 GUIDir.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <time.h>
#include <signal.h>
#include <gtk/gtk.h>
#include <gdk/gdkrgb.h>
#include "GUIDir.hh"
```

Include dependency graph for GUIDir.cc:



Functions

- **G_LOCK_DEFINE_STATIC** (gmInputMutex)
- int **main** (int argc, char *argv[ ])
- gint **WrapOnDelete** (GtkWidget *gwSender, GdkEventAny *geaEvent)
- void **WrapOnHideToggled** (GtkToggleButton *gtbSender, gpointer gpData)
- gint **WrapOnConnectClick** (GtkWidget *gwSender, gpointer gpData)
- void **WrapOnFreezeToggled** (GtkToggleButton *gtbSender, gpointer gpData)
- gint **WrapOnExposeWorm** (GtkWidget *gwSender, GdkEventExpose *geeEvent)
- gint **WrapOnMotionWorm** (GtkWidget *gwSender, GdkEventMotion *gemEvent)
- gint **WrapOnPaletteActivate** (GtkWidget *gwSender, gpointer gpData)
- void **WrapOnSaveClicks** (GtkWidget *gwSender, gpointer gpData)
- void **WrapOnGdkInput** (gpointer gpData, gint giSource, GdkInputCondition gicCondition)

Variables

- const char * **cpWindowTxt** = "Direction"
- const char * **cpLServerTxt** = "Server"

- const char * **cpBConnectTxt** = "Connect"
- const char * **cpBDisconnectTxt** = "Disconnect"
- const char * **cpCBFreezeTxt** = "Freeze"
- const char * **cpLAlgorithmTxt** = "Algorithm"
- const char * **cpaLAlgorithmMenu** []
- const char * **cpLSoundSpeedTxt** = "Sound speed"
- const char * **cpLLowFrequencyLimitTxt** = "Low frequency limit"
- const char * **cpLIntegrationTimeTxt** = "Integration time"
- const char * **cpLScalingTxt** = "Scaling"
- const char * **cpaLScalingMenu** []
- const char * **cpLScalingExponentTxt** = "Scaling exponent"
- const char * **cpLRemoveNoiseTxt** = "Remove noise"
- const char * **cpaLRemoveNoiseMenu** []
- const char * **cpLAlphaTxt** = "Alpha"
- const char * **cpLMeanLengthTxt** = "Mean length"
- const char * **cpLGapLengthTxt** = "Gap length"
- const char * **cpCBNormalizeTxt** = "Normalize"
- const char * **cpCBNoFilterTxt** = "No filter"
- const char * **cpLPaletteTxt** = "Palette"
- const char * **cpaLPaletteMenu** []
- const char * **cpLLeftDirectionTxt** = "Left direction"
- const char * **cpLRightDirectionTxt** = "Right direction"
- const char * **cpLSectorCountTxt** = "Sector count"
- const char * **cpLDirectionScaleTxt** = "Direction scale"
- const char * **cpCBSavingTxt** = "Saving"
- const char * **cpBSaveTxt** = "Save"
- const char * **cpFSSaveTxt** = "Save to TIFF file"
- **clGUIDir * GUIDir**

8.59.1 Function Documentation

8.59.1.1 G_LOCK_DEFINE_STATIC (gmInputMutex)

8.59.1.2 int main (int *argc*, char * *argv*[])

Definition at line 79 of file GUIDir.cc.

References clGUIDir::Exec(), and GUIDir.

8.59.1.3 gint WrapOnDelete (GtkWidget * *gwSender*, GdkEventAny * *geaEvent*)

Definition at line 92 of file GUIDir.cc.

References GUIDir, and clGUIDir::OnDelete().

8.59.1.4 void WrapOnHideToggled (GtkToggleButton ∗ *gtbSender*, gpointer *gpData*)

Definition at line 98 of file GUIDir.cc.

References GUIDir, and clGUIDir::OnHideToggled().

Referenced by clSpectGUI::ConnectSignals(), clGUILofar::ConnectSignals(), and clGUIDir::ConnectSignals().

8.59.1.5 gint WrapOnConnectClick (GtkWidget ∗ *gwSender*, gpointer *gpData*)

Definition at line 104 of file GUIDir.cc.

References GUIDir, and clGUIDir::OnConnectClick().

8.59.1.6 void WrapOnFreezeToggled (GtkToggleButton ∗ *gtbSender*, gpointer *gpData*)

Definition at line 110 of file GUIDir.cc.

References GUIDir, and clGUIDir::OnFreezeToggled().

Referenced by clSpectGUI::ConnectSignals(), clGUILofar::ConnectSignals(), and clGUIDir::ConnectSignals().

8.59.1.7 gint WrapOnExposeWorm (GtkWidget ∗ *gwSender*, GdkEventExpose ∗ *geeEvent*)

Definition at line 116 of file GUIDir.cc.

References GUIDir, and clGUIDir::OnExposeWorm().

Referenced by clGUIDir::ConnectSignals().

8.59.1.8 gint WrapOnMotionWorm (GtkWidget ∗ *gwSender*, GdkEventMotion ∗ *gemEvent*)

Definition at line 122 of file GUIDir.cc.

References GUIDir, and clGUIDir::OnMotionWorm().

Referenced by clGUIDir::ConnectSignals().

8.59.1.9 gint WrapOnPaletteActivate (GtkWidget ∗ *gwSender*, gpointer *gpData*)

Definition at line 128 of file GUIDir.cc.

References GUIDir, and clGUIDir::OnPaletteActivate().

Referenced by clSpectGUI::ConnectSignals(), clGUILofar::ConnectSignals(), clGUILocate::ConnectSignals(), and clGUIDir::ConnectSignals().

8.59.1.10 void WrapOnSaveClicks (GtkWidget * *gwSender*, gpointer *gpData*)

Definition at line 134 of file GUIDir.cc.

References GUIDir, and clGUIDir::OnSaveClicks().

Referenced by clSpectGUI::ConnectSignals(), clGUILofar::ConnectSignals(), and clGUIDir::ConnectSignals().

8.59.1.11 void WrapOnGdkInput (gpointer *gpData*, gint *giSource*, GdkInputCondition *gicCondition*)

Definition at line 140 of file GUIDir.cc.

References GUIDir, and clGUIDir::OnGdkInput().

8.59.2 Variable Documentation

8.59.2.1 const char* cpWindowTxt = "Direction" `[static]`

Definition at line 37 of file GUIDir.cc.

Referenced by clGUIDir::Build(), and clGUIDir::clGUIDir().

8.59.2.2 const char* cpLServerTxt = "Server" `[static]`

Definition at line 39 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable1().

8.59.2.3 const char* cpBConnectTxt = "Connect" `[static]`

Definition at line 40 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable1().

8.59.2.4 const char* cpBDisconnectTxt = "Disconnect" `[static]`

Definition at line 41 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable1().

8.59.2.5 const char* cpCBFreezeTxt = "Freeze" `[static]`

Definition at line 42 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable1().

8.59.2.6 const char* cpLAlgorithmTxt = "Algorithm" [static]

Definition at line 44 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable2().

8.59.2.7 const char* cpaLAlgorithmMenu[] [static]

Initial value:

```
{ "Beamforming", "Correlation",
   "Beamforming & Correlation", "Spectrum power/phase" }
```

Definition at line 45 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable2().

8.59.2.8 const char* cpLSoundSpeedTxt = "Sound speed" [static]

Definition at line 47 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable2().

8.59.2.9 const char* cpLLowFrequencyLimitTxt = "Low frequency limit" [static]

Definition at line 48 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable2().

8.59.2.10 const char* cpLIntegrationTimeTxt = "Integration time" [static]

Definition at line 49 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable2().

8.59.2.11 const char* cpLScalingTxt = "Scaling" [static]

Definition at line 51 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.12 const char* cpaLScalingMenu[] [static]

Initial value:

```
{ "Linear", "Logarithmic",
   "Exponential", "Sine" }
```

Definition at line 52 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.13 const char* cpLScalingExponentTxt = "Scaling exponent" [static]

Definition at line 54 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.14 const char* cpLRemoveNoiseTxt = "Remove noise" [static]

Definition at line 55 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.15 const char* cpaLRemoveNoiseMenu[] [static]

Initial value:

```
{ "None", "TPSW", "OTA",
   "Diff", "InvDiff" }
```

Definition at line 56 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.16 const char* cpLAlphaTxt = "Alpha" [static]

Definition at line 58 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.17 const char* cpLMeanLengthTxt = "Mean length" [static]

Definition at line 59 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.18 const char∗ cpLGapLengthTxt = "Gap length" [static]

Definition at line 60 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.19 const char∗ cpCBNormalizeTxt = "Normalize" [static]

Definition at line 61 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.20 const char∗ cpCBNoFilterTxt = "No filter" [static]

Definition at line 62 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.21 const char∗ cpLPaletteTxt = "Palette" [static]

Definition at line 63 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.22 const char∗ cpaLPaletteMenu[] [static]

Initial value:

```
{ "BW", "HSV", "Light", "Temp",
   "Dir", "Green", "Green2", "PureGreen", "WB" }
```

Definition at line 64 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable3().

8.59.2.23 const char∗ cpLLeftDirectionTxt = "Left direction" [static]

Definition at line 67 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable4().

8.59.2.24 const char∗ cpLRightDirectionTxt = "Right direction" [static]

Definition at line 68 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable4().

8.59.2.25 const char* cpLSectorCountTxt = "Sector count" [static]

Definition at line 69 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable4().

8.59.2.26 const char* cpLDirectionScaleTxt = "Direction scale" [static]

Definition at line 70 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable4().

8.59.2.27 const char* cpCBSavingTxt = "Saving" [static]

Definition at line 71 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable4().

8.59.2.28 const char* cpBSaveTxt = "Save" [static]

Definition at line 72 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable4().

8.59.2.29 const char* cpFSSaveTxt = "Save to TIFF file" [static]

Definition at line 73 of file GUIDir.cc.

Referenced by clGUIDir::BuildTable4().

8.59.2.30 clGUIDir* GUIDir

Definition at line 76 of file GUIDir.cc.

Referenced by main(), WrapOnConnectClick(), WrapOnDelete(), WrapOnExposeWorm(), WrapOnFreezeToggled(), WrapOnGdkInput(), WrapOnHideToggled(), WrapOnMotionWorm(), WrapOnPaletteActivate(), and WrapOnSaveClicks().

8.60 GUIDir.hh File Reference

`#include <ctime>`

`#include <string>`

`#include <Alloc.hh>`

`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

`#include "CfgFile.hh"`

`#include "FrameBuf.hh"`

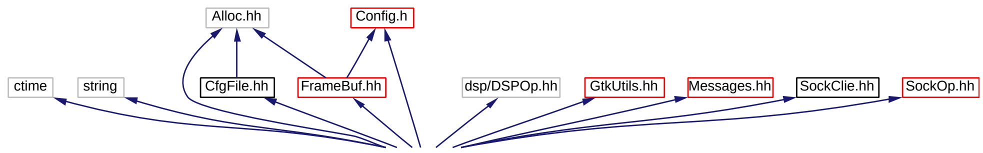
`#include "GtkUtils.hh"`

`#include "Messages.hh"`

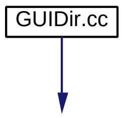
`#include "SockClie.hh"`

`#include "SockOp.hh"`

Include dependency graph for GUIDir.hh:



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



Compounds

- class **clGUIDir**

  Direction finding GUI.

Defines

- #define **DGUI_VER_MAJ** GLOBAL_VERSMAJ
- #define **DGUI_VER_MIN** GLOBAL_VERSMIN

- #define **DGUI_VER_PL** GLOBAL_VERSPL
- #define **DGUI_PADDING** 8
- #define **DGUI_ENTRY_WIDTH** 80
- #define **DGUI_SERVER_MAXLEN** 255
- #define **DGUI_WORM_BG** 0x00ffffff
- #define **DGUI_WORM_FG** 0x00000000
- #define **DGUI_CONV_BUF_SIZE** 255
- #define **DGUI_ALGORITHM_ITEMS** 4
- #define **DGUI_SCALING_ITEMS** 4
- #define **DGUI_REMOVE_NOISE_ITEMS** 5
- #define **DGUI_PALETTE_ITEMS** 9

Enumerations

- enum {

  DGUI_PAL_BW = 0, **DGUI_PAL_HSV** = 1, **DGUI_PAL_LIGHT** = 2, **DGUI_PAL_TEMP** = 3,

  DGUI_PAL_DIR = 4, **DGUI_PAL_GREEN** = 5, **DGUI_PAL_-GREEN2** = 6, **DGUI_PAL_PUREGREEN** = 7,

  DGUI_PAL_WB = 8 }

  Available palettes.

8.60.1 Define Documentation

8.60.1.1 #define DGUI_VER_MAJ GLOBAL_VERSMAJ

Definition at line 41 of file GUIDir.hh.

Referenced by clGUIDir::clGUIDir().

8.60.1.2 #define DGUI_VER_MIN GLOBAL_VERSMIN

Definition at line 42 of file GUIDir.hh.

Referenced by clGUIDir::clGUIDir().

8.60.1.3 #define DGUI_VER_PL GLOBAL_VERSPL

Definition at line 43 of file GUIDir.hh.

Referenced by clGUIDir::clGUIDir().

8.60.1.4 #define DGUI_PADDING 8

Definition at line 44 of file GUIDir.hh.

Referenced by clGUIDir::Build(), clGUIDir::BuildTable1(), clGUIDir::BuildTable2(), clGUIDir::BuildTable3(), and clGUIDir::BuildTable4().

8.60.1.5 #define DGUI_ENTRY_WIDTH 80

Definition at line 45 of file GUIDir.hh.

Referenced by clGUIDir::BuildTable2(), clGUIDir::BuildTable3(), and clGUIDir::BuildTable4().

8.60.1.6 #define DGUI_SERVER_MAXLEN 255

Definition at line 46 of file GUIDir.hh.

Referenced by clGUIDir::BuildTable1(), clGUIDir::OnConnectClick(), and clGUIDir::ParseServerStr().

8.60.1.7 #define DGUI_WORM_BG 0x00ffffff

Definition at line 47 of file GUIDir.hh.

Referenced by clGUIDir::BuildDrawingPrims().

8.60.1.8 #define DGUI_WORM_FG 0x00000000

Definition at line 48 of file GUIDir.hh.

Referenced by clGUIDir::BuildDrawingPrims().

8.60.1.9 #define DGUI_CONV_BUF_SIZE 255

Definition at line 49 of file GUIDir.hh.

Referenced by clGUIDir::BuildTable2(), clGUIDir::BuildTable4(), clGUIDir::OnGdkInput(), and clGUIDir::PrintStatus().

8.60.1.10 #define DGUI_ALGORITHM_ITEMS 4

Definition at line 51 of file GUIDir.hh.

Referenced by clGUIDir::BuildTable2(), and clGUIDir::GetSettings().

8.60.1.11 #define DGUI_SCALING_ITEMS 4

Definition at line 52 of file GUIDir.hh.

Referenced by clGUIDir::BuildTable3(), and clGUIDir::GetSettings().

8.60.1.12 #define DGUI_REMOVE_NOISE_ITEMS 5

Definition at line 53 of file GUIDir.hh.

Referenced by clGUIDir::BuildTable3(), and clGUIDir::GetSettings().

8.60.1.13 #define DGUI_PALETTE_ITEMS 9

Definition at line 54 of file GUIDir.hh.

Referenced by clGUIDir::BuildTable3(), and clGUIDir::ConnectSignals().

8.60.2 Enumeration Type Documentation

8.60.2.1 anonymous enum

Available palettes.

Enumeration values:

- **DGUI_PAL_BW**
- **DGUI_PAL_HSV**
- **DGUI_PAL_LIGHT**
- **DGUI_PAL_TEMP**
- **DGUI_PAL_DIR**
- **DGUI_PAL_GREEN**
- **DGUI_PAL_GREEN2**
- **DGUI_PAL_PUREGREEN**
- **DGUI_PAL_WB**

Definition at line 60 of file GUIDir.hh.

8.61 GUILevel.cc File Reference

`#include <stdlib.h>`

`#include <string.h>`

`#include <signal.h>`

`#include <unistd.h>`

`#include <math.h>`

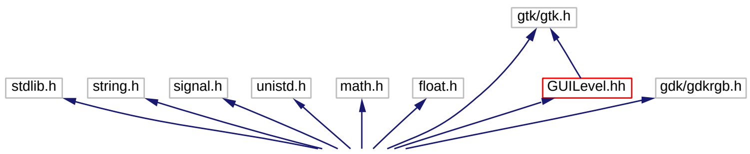
`#include <float.h>`

`#include <gtk/gtk.h>`

`#include <gdk/gdkrgb.h>`

`#include "GUILevel.hh"`

Include dependency graph for GUILevel.cc:



Functions

- **G_LOCK_DEFINE_STATIC** (gmInputMutex)
- int **main** (int argc, char *argv[ ])
- gboolean **WrapOnDelete** (GtkWidget *gwSender, GdkEvent *geEvent, gpointer gpData)
- void **WrapOnConnectClick** (GtkButton *gbSender, gpointer gpData)
- gboolean **WrapOnSofarExpose** (GtkWidget *gwSender, GdkEventExpose *geeEvent, gpointer gpData)
- gboolean **WrapOnSofarMotion** (GtkWidget *gwSender, GdkEventMotion *gemEvent, gpointer gpData)
- gboolean **WrapOnSofarConfigure** (GtkWidget *gwSender, GdkEventConfigure *gecEvent, gpointer gpData)
- void **WrapOnGdkInput** (gpointer gpData, gint iSource, GdkInputCondition gicCondition)

Variables

- const char * **cpWindowTxt** = "Level"
- const char * **cpLServerTxt** = "Server"
- const char * **cpaLChannelTxt** [ ] = { "Channel", "Direction" }
- const char * **cpBConnectTxt** = "Connect"

- const char * **cpLAlgorithmTxt** = "Algorithm"
- const char * **cpaLAlgorithmMenu** [ ]
- const char * **cpLIntegrationTimeTxt** = "Integration time"
- const char * **cpLLowFrequencyTxt** = "Lower frequency"
- const char * **cpLHighFrequencyTxt** = "Higher frequency"
- const char * **cpLDisplayLowTxt** = "Display low limit"
- const char * **cpLDisplayHighTxt** = "Display high limit"
- **clGUILevel GUILevel**

8.61.1 Function Documentation

8.61.1.1 G_LOCK_DEFINE_STATIC (gmInputMutex)

8.61.1.2 int main (int *argc*, char * *argv*[ ])

Definition at line 55 of file GUILevel.cc.

References GUILevel, and clGUILevel::Main().

8.61.1.3 gboolean WrapOnDelete (GtkWidget * *gwSender*, GdkEvent * *geEvent*, gpointer *gpData*)

Definition at line 63 of file GUILevel.cc.

References GUILevel, and clGUILevel::OnDelete().

8.61.1.4 void WrapOnConnectClick (GtkButton * *gbSender*, gpointer *gpData*)

Definition at line 70 of file GUILevel.cc.

References GUILevel, and clGUILevel::OnConnectClick().

8.61.1.5 gboolean WrapOnSofarExpose (GtkWidget * *gwSender*, GdkEventExpose * *geeEvent*, gpointer *gpData*)

Definition at line 76 of file GUILevel.cc.

References GUILevel, and clGUILevel::OnSofarExpose().

Referenced by clGUILevel::ConnectSignals().

8.61.1.6 gboolean WrapOnSofarMotion (GtkWidget * *gwSender*, GdkEventMotion * *gemEvent*, gpointer *gpData*)

Definition at line 83 of file GUILevel.cc.

References GUILevel, and clGUILevel::OnSofarMotion().

Referenced by clGUILevel::ConnectSignals().

8.61.1.7 gboolean WrapOnSofarConfigure (GtkWidget * *gwSender*, GdkEventConfigure * *gecEvent*, gpointer *gpData*)

Definition at line 90 of file GUILevel.cc.

References GUILevel, and clGUILevel::OnSofarConfigure().

Referenced by clGUILevel::ConnectSignals().

8.61.1.8 void WrapOnGdkInput (gpointer *gpData*, gint *iSource*, GdkInputCondition *gicCondition*)

Definition at line 97 of file GUILevel.cc.

References GUILevel, and clGUILevel::OnGdkInput().

8.61.2 Variable Documentation

8.61.2.1 const char* cpWindowTxt = "Level" [static]

Definition at line 37 of file GUILevel.cc.

Referenced by clGUILevel::Build(), and clGUILevel::Main().

8.61.2.2 const char* cpLServerTxt = "Server" [static]

Definition at line 39 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable1().

8.61.2.3 const char* cpaLChannelTxt[] = { "Channel", "Direction" } [static]

Definition at line 40 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable1().

8.61.2.4 const char* cpBConnectTxt = "Connect" [static]

Definition at line 41 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable1().

8.61.2.5 const char* cpLAlgorithmTxt = "Algorithm" [static]

Definition at line 43 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable2().

8.61.2.6 const char* cpaLAlgorithmMenu[] [static]

Initial value:

```
{ "Peak", "RMS", "Mean", "Median",
   "StdDev" }
```

Definition at line 44 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable2().

8.61.2.7 const char* cpLIntegrationTimeTxt = "Integration time" [static]

Definition at line 46 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable2().

8.61.2.8 const char* cpLLowFrequencyTxt = "Lower frequency" [static]

Definition at line 47 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable2().

8.61.2.9 const char* cpLHighFrequencyTxt = "Higher frequency" [static]

Definition at line 48 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable2().

8.61.2.10 const char* cpLDisplayLowTxt = "Display low limit" [static]

Definition at line 49 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable2().

8.61.2.11 const char* cpLDisplayHighTxt = "Display high limit" [static]

Definition at line 50 of file GUILevel.cc.

Referenced by clGUILevel::BuildTable2().

8.61.2.12 clGUILevel GUILevel

Definition at line 52 of file GUILevel.cc.

Referenced by main(), WrapOnConnectClick(), WrapOnDelete(), WrapOnGdkInput(), WrapOnSofarConfigure(), WrapOnSofarExpose(), and WrapOnSofarMotion().

8.62 GUILevel.hh File Reference

`#include <gtk/gtk.h>`

`#include <Alloc.hh>`

`#include "Config.h"`

`#include "CfgFile.hh"`

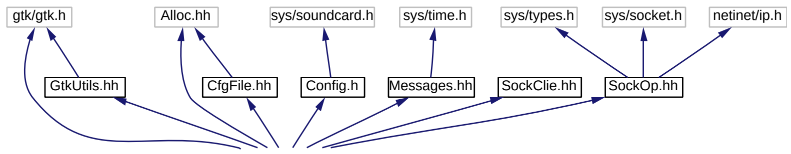
`#include "GtkUtils.hh"`

`#include "Messages.hh"`

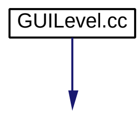
`#include "SockClie.hh"`

`#include "SockOp.hh"`

Include dependency graph for GUILevel.hh:



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



Compounds

- class **clGUILevel**

  GUI for level server.

Defines

- #define **GUILEV_VER_MAJ** GLOBAL_VERSMAJ
- #define **GUILEV_VER_MIN** GLOBAL_VERSMIN
- #define **GUILEV_VER_PL** GLOBAL_VERSPL
- #define **GUILEV_WSPACING** 8
- #define **GUILEV_ENTRY_WIDTH** 80
- #define **GUILEV_SOFAR_WIDTH** 600

- #define **GUILEV_SOFAR_HEIGHT** 100
- #define **GUILEV_SERVER_MAXLEN** 256
- #define **GUILEV_CH_LOWER** 1
- #define **GUILEV_CH_UPPER** 16
- #define **GUILEV_LINE_BG** 0x00ffffff
- #define **GUILEV_LINE_FG** 0x000000ff
- #define **GUILEV_ALGORITHM_ITEMS** 5

8.62.1 Define Documentation

8.62.1.1 #define GUILEV_VER_MAJ GLOBAL_VERSMAJ

Definition at line 38 of file GUILevel.hh.

Referenced by clGUILevel::Main().

8.62.1.2 #define GUILEV_VER_MIN GLOBAL_VERSMIN

Definition at line 39 of file GUILevel.hh.

Referenced by clGUILevel::Main().

8.62.1.3 #define GUILEV_VER_PL GLOBAL_VERSPL

Definition at line 40 of file GUILevel.hh.

Referenced by clGUILevel::Main().

8.62.1.4 #define GUILEV_WSPACING 8

Definition at line 41 of file GUILevel.hh.

Referenced by clGUILevel::Build(), clGUILevel::BuildTable1(), and clGUILevel::BuildTable2().

8.62.1.5 #define GUILEV_ENTRY_WIDTH 80

Definition at line 42 of file GUILevel.hh.

Referenced by clGUILevel::BuildTable2().

8.62.1.6 #define GUILEV_SOFAR_WIDTH 600

Definition at line 43 of file GUILevel.hh.

Referenced by clGUILevel::BuildSofar().

8.62.1.7 #define GUILEV_SOFAR_HEIGHT 100

Definition at line 44 of file GUILevel.hh.

Referenced by clGUILevel::BuildSofar().

8.62.1.8 #define GUILEV_SERVER_MAXLEN 256

Definition at line 45 of file GUILevel.hh.

Referenced by clGUILevel::BuildTable1().

8.62.1.9 #define GUILEV_CH_LOWER 1

Definition at line 46 of file GUILevel.hh.

Referenced by clGUILevel::BuildTable1().

8.62.1.10 #define GUILEV_CH_UPPER 16

Definition at line 47 of file GUILevel.hh.

Referenced by clGUILevel::BuildTable1().

8.62.1.11 #define GUILEV_LINE_BG 0x00ffffff

Definition at line 48 of file GUILevel.hh.

Referenced by clGUILevel::BuildDrawingPrims().

8.62.1.12 #define GUILEV_LINE_FG 0x000000ff

Definition at line 49 of file GUILevel.hh.

Referenced by clGUILevel::BuildDrawingPrims().

8.62.1.13 #define GUILEV_ALGORITHM_ITEMS 5

Definition at line 51 of file GUILevel.hh.

Referenced by clGUILevel::BuildTable2(), and clGUILevel::SendSettings().

8.63 GUILocate.cc File Reference

`#include <stdlib.h>`

`#include <string.h>`

`#include <signal.h>`

`#include <unistd.h>`

`#include <math.h>`

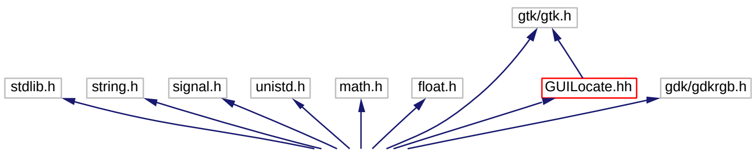
`#include <float.h>`

`#include <gtk/gtk.h>`

`#include <gdk/gdkrgb.h>`

`#include "GUILocate.hh"`

Include dependency graph for GUILocate.cc:



Functions

- **G_LOCK_DEFINE_STATIC** (gmInputMutex)
- int **main** (int argc, char *argv[ ])
- gboolean **WrapOnDelete** (GtkWidget *gwSender, GdkEvent *geEvent, gpointer gpData)
- void **WrapOnConnectClick** (GtkButton *gbSender, gpointer gpData)
- void **WrapOnPaletteActivate** (GtkMenuItem *gmiSender, gpointer gpData)
- gboolean **WrapOnLocateExpose** (GtkWidget *gwSender, GdkEventExpose *geeEvent, gpointer gpData)
- gboolean **WrapOnLocateMotion** (GtkWidget *gwSender, GdkEventMotion *gemEvent, gpointer gpData)
- void **WrapOnGdkInput** (gpointer gpData, gint iSource, GdkInputCondition gicCondition)

Variables

- const char * **cpWindowTxt** = "Locate"
- const char * **cpLServerTxt** = "Server"
- const char * **cpBConnectTxt** = "Connect"
- const char * **cpLPaletteTxt** = "Palette"

- const char * **cpaLPaletteMenu** []
- **clGUILocate GUILocate**

8.63.1 Function Documentation

8.63.1.1 G_LOCK_DEFINE_STATIC (gmInputMutex)

8.63.1.2 int main (int *argc*, char * *argv*[ ])

Definition at line 50 of file GUILocate.cc.

References GUILocate, and clGUILocate::Main().

8.63.1.3 gboolean WrapOnDelete (GtkWidget * *gwSender*, GdkEvent * *geEvent*, gpointer *gpData*)

Definition at line 58 of file GUILocate.cc.

References GUILocate, and clGUILocate::OnDelete().

8.63.1.4 void WrapOnConnectClick (GtkButton * *gbSender*, gpointer *gpData*)

Definition at line 65 of file GUILocate.cc.

References GUILocate, and clGUILocate::OnConnectClick().

8.63.1.5 void WrapOnPaletteActivate (GtkMenuItem * *gmiSender*, gpointer *gpData*)

Definition at line 71 of file GUILocate.cc.

References GUILocate, and clGUILocate::OnPaletteActivate().

8.63.1.6 gboolean WrapOnLocateExpose (GtkWidget * *gwSender*, GdkEventExpose * *geeEvent*, gpointer *gpData*)

Definition at line 77 of file GUILocate.cc.

References GUILocate, and clGUILocate::OnLocateExpose().

Referenced by clGUILocate::ConnectSignals().

8.63.1.7 gboolean WrapOnLocateMotion (GtkWidget * *gwSender*, GdkEventMotion * *gemEvent*, gpointer *gpData*)

Definition at line 84 of file GUILocate.cc.

References GUILocate, and clGUILocate::OnLocateMotion().

Referenced by clGUILocate::ConnectSignals().

8.63.1.8 void WrapOnGdkInput (gpointer *gpData*, gint *iSource*, GdkInputCondition *gicCondition*)

Definition at line 91 of file GUILocate.cc.

References GUILocate, and clGUILocate::OnGdkInput().

8.63.2 Variable Documentation

8.63.2.1 const char* cpWindowTxt = "Locate" `[static]`

Definition at line 38 of file GUILocate.cc.

Referenced by clGUILocate::Build(), and clGUILocate::Main().

8.63.2.2 const char* cpLServerTxt = "Server" `[static]`

Definition at line 40 of file GUILocate.cc.

Referenced by clGUILocate::BuildTable1().

8.63.2.3 const char* cpBConnectTxt = "Connect" `[static]`

Definition at line 41 of file GUILocate.cc.

Referenced by clGUILocate::BuildTable1().

8.63.2.4 const char* cpLPaletteTxt = "Palette" `[static]`

Definition at line 43 of file GUILocate.cc.

Referenced by clGUILocate::BuildTable2().

8.63.2.5 const char* cpaLPaletteMenu[] `[static]`

Initial value:

```
{ "BW", "HSV", "Light", "Temp",
   "Dir", "Green", "Green2", "PureGreen", "WB" }
```

Definition at line 44 of file GUILocate.cc.

Referenced by clGUILocate::BuildTable2().

8.63.2.6 clGUILocate GUILocate

Definition at line 47 of file GUILocate.cc.

Referenced by main(), WrapOnConnectClick(), WrapOnDelete(), WrapOnGdkInput(), WrapOnLocateExpose(), WrapOnLocateMotion(), and WrapOnPaletteActivate().

8.64 GUILocate.hh File Reference

`#include <gtk/gtk.h>`

`#include <Alloc.hh>`

`#include "Config.h"`

`#include "CfgFile.hh"`

`#include "GtkUtils.hh"`

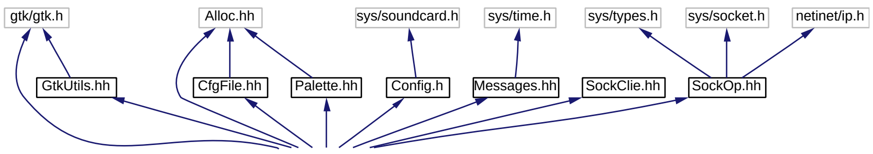
`#include "Messages.hh"`

`#include "Palette.hh"`

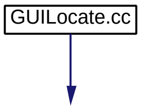
`#include "SockClie.hh"`

`#include "SockOp.hh"`

Include dependency graph for GUILocate.hh:



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



Compounds

- class **clGUILocate**

  GUI for locating.

Defines

- #define **GUILOC_VER_MAJ** GLOBAL_VERSMAJ
- #define **GUILOC_VER_MIN** GLOBAL_VERSMIN
- #define **GUILOC_VER_PL** GLOBAL_VERSPL
- #define **GUILOC_WSPACING** 8
- #define **GUILOC_ENTRY_WIDTH** 80

- #define **GUILOC_SERVER_MAXLEN** 256
- #define **GUILOC_LOCATE_BG** 0x00000000
- #define **GUILOC_LOCATE_FG** 0x00ffffff
- #define **GUILOC_PALETTE_ITEMS** 9

Enumerations

- enum {
  GUILOC_PAL_BW = 0, **GUILOC_PAL_HSV** = 1, **GUILOC_PAL_LIGHT** = 2, **GUILOC_PAL_TEMP** = 3,
  GUILOC_PAL_DIR = 4, **GUILOC_PAL_GREEN** = 5, **GUILOC_PAL_GREEN2** = 6, **GUILOC_PAL_PUREGREEN** = 7,
  GUILOC_PAL_WB = 8 }

  Available palettes.

8.64.1 Define Documentation

8.64.1.1 #define GUILOC_VER_MAJ GLOBAL_VERSMAJ

Definition at line 39 of file GUILocate.hh.

Referenced by clGUILocate::Main().

8.64.1.2 #define GUILOC_VER_MIN GLOBAL_VERSMIN

Definition at line 40 of file GUILocate.hh.

Referenced by clGUILocate::Main().

8.64.1.3 #define GUILOC_VER_PL GLOBAL_VERSPL

Definition at line 41 of file GUILocate.hh.

Referenced by clGUILocate::Main().

8.64.1.4 #define GUILOC_WSPACING 8

Definition at line 42 of file GUILocate.hh.

Referenced by clGUILocate::Build(), clGUILocate::BuildTable1(), and clGUILocate::BuildTable2().

8.64.1.5 #define GUILOC_ENTRY_WIDTH 80

Definition at line 43 of file GUILocate.hh.

8.64.1.6 #define GUILOC_SERVER_MAXLEN 256

Definition at line 44 of file GUILocate.hh.

Referenced by clGUILocate::BuildTable1().

8.64.1.7 #define GUILOC_LOCATE_BG 0x00000000

Definition at line 45 of file GUILocate.hh.

Referenced by clGUILocate::BuildDrawingPrims().

8.64.1.8 #define GUILOC_LOCATE_FG 0x00ffffff

Definition at line 46 of file GUILocate.hh.

Referenced by clGUILocate::BuildDrawingPrims().

8.64.1.9 #define GUILOC_PALETTE_ITEMS 9

Definition at line 48 of file GUILocate.hh.

Referenced by clGUILocate::BuildTable2(), and clGUILocate::ConnectSignals().

8.64.2 Enumeration Type Documentation

8.64.2.1 anonymous enum

Available palettes.

Enumeration values:

GUILOC_PAL_BW

GUILOC_PAL_HSV

GUILOC_PAL_LIGHT

GUILOC_PAL_TEMP

GUILOC_PAL_DIR

GUILOC_PAL_GREEN

GUILOC_PAL_GREEN2

GUILOC_PAL_PUREGREEN

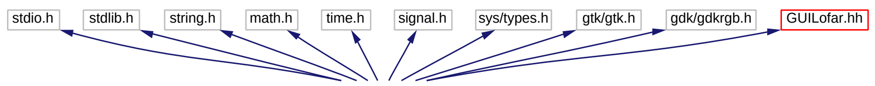
GUILOC_PAL_WB

Definition at line 54 of file GUILocate.hh.

8.65 GUILofar.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <time.h>
#include <signal.h>
#include <sys/types.h>
#include <gtk/gtk.h>
#include <gdk/gdkrgb.h>
#include "GUILofar.hh"
```

Include dependency graph for GUILofar.cc:



Functions

- **G_LOCK_DEFINE_STATIC** (gmInputMutex)
- int **main** (int argc, char *argv[])
- gint **WrapOnDelete** (GtkWidget *gwSender, GdkEventAny *geaEvent)
- void **WrapOnHideToggled** (GtkToggleButton *gtbSender, gpointer gpData)
- gint **WrapOnConnectClick** (GtkWidget *gwSender, gpointer gpData)
- void **WrapOnFreezeToggled** (GtkToggleButton *gtbSender, gpointer gpData)
- gint **WrapOnExposeLofar** (GtkWidget *gwSender, GdkEventExpose *geeEvent, gpointer gpData)
- gint **WrapOnConfigureLofar** (GtkWidget *gwSender, GdkEventConfigure *gecEvent, gpointer gpData)
- gint **WrapOnExposeLine** (GtkWidget *gwSender, GdkEventExpose *geeEvent, gpointer gpData)
- gint **WrapOnExposeCursor** (GtkWidget *gwSender, GdkEventExpose *geeEvent, gpointer gpData)
- gint **WrapOnMotionLofar** (GtkWidget *gwSender, GdkEventMotion *gemEvent, gpointer gpData)
- gint **WrapOnPaletteActivate** (GtkWidget *gwSender, gpointer gpData)
- void **WrapOnAverageToggled** (GtkToggleButton *gtbSender, gpointer gpData)

- void **WrapOnClipValueChanged** (GtkAdjustment *gaSender, gpointer gpData)
- void **WrapOnSaveClicks** (GtkWidget *gwSender, gpointer gpData)
- void **WrapOnGdkInput** (gpointer gpData, gint giSource, GdkInputCondition gicCondition)

Variables

- const char * **cpWindowTxt** = "LOFAR / DEMON"
- const char * **cpLServerTxt** = "Server"
- const char * **cpaLChannelTxt** [ ] = { "Channel", "Direction" }
- const char * **cpBConnectTxt** = "Connect"
- const char * **cpBDisconnectTxt** = "Disconnect"
- const char * **cpCBFreezeTxt** = "Freeze"
- const char * **cpLTypeTxt** = "Type"
- const char * **cpaLTypeMenu** [ ] = { "FFT", "Cepstrum", "Autocorr." }
- const char * **cpLWindowTxt** = "Window"
- const char * **cpaLWindowMenu** [ ]
- const char * **cpLWinParamTxt** = "Window param."
- const char * **cpLWinLengthTxt** = "Window length"
- const char * **cpaLWinLengthMenu** [ ]
- const char * **cpLLowerFreqTxt** = "Lower freq."
- const char * **cpLHigherFreqTxt** = "Higher freq."
- const char * **cpLOverlapTxt** = "Overlap %"
- const char * **cpLRemoveNoiseTxt** = "Remove noise"
- const char * **cpaLRemoveNoiseMenu** [ ]
- const char * **cpLAlphaTxt** = "Alpha"
- const char * **cpLMeanLengthTxt** = "Mean length"
- const char * **cpLGapLengthTxt** = "Gap length"
- const char * **cpLAverageCountTxt** = "Average count"
- const char * **cpLClipTxt** = "Clip"
- const char * **cpaLClipMenu** [ ]
- const char * **cpCBLinearTxt** = "Linear"
- const char * **cpCBDemonTxt** = "DEMON"
- const char * **cpLPaletteTxt** = "Palette"
- const char * **cpaLPaletteMenu** [ ]
- const char * **cpCBAverageTxt** = "ContAvg"
- const char * **cpCBSavingTxt** = "Saving"
- const char * **cpBSaveTxt** = "Save"
- const char * **cpFSSaveTxt** = "Save to TIFF file"
- const char * **cpLClipValueTxt** = "Clip value"
- **clGUILofar** * **GUILofar**

8.65.1 Function Documentation

8.65.1.1 G_LOCK_DEFINE_STATIC (gmInputMutex)

8.65.1.2 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 86 of file GUILofar.cc.

References clGUILofar::Exec(), and GUILofar.

8.65.1.3 gint WrapOnDelete (GtkWidget ∗ *gwSender*, GdkEventAny ∗ *geaEvent*)

Definition at line 99 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnDelete().

8.65.1.4 void WrapOnHideToggled (GtkToggleButton ∗ *gtbSender*, gpointer *gpData*)

Definition at line 105 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnHideToggled().

8.65.1.5 gint WrapOnConnectClick (GtkWidget ∗ *gwSender*, gpointer *gpData*)

Definition at line 111 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnConnectClick().

8.65.1.6 void WrapOnFreezeToggled (GtkToggleButton ∗ *gtbSender*, gpointer *gpData*)

Definition at line 117 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnFreezeToggled().

8.65.1.7 gint WrapOnExposeLofar (GtkWidget ∗ *gwSender*, GdkEventExpose ∗ *geeEvent*, gpointer *gpData*)

Definition at line 123 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnExposeLofar().

Referenced by clGUILofar::ConnectSignals().

8.65.1.8 gint WrapOnConfigureLofar (GtkWidget * *gwSender*, GdkEventConfigure * *gecEvent*, gpointer *gpData*)

Definition at line 130 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnConfigureLofar().

Referenced by clGUILofar::ConnectSignals().

8.65.1.9 gint WrapOnExposeLine (GtkWidget * *gwSender*, GdkEventExpose * *geeEvent*, gpointer *gpData*)

Definition at line 137 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnExposeLine().

Referenced by clGUILofar::ConnectSignals().

8.65.1.10 gint WrapOnExposeCursor (GtkWidget * *gwSender*, GdkEventExpose * *geeEvent*, gpointer *gpData*)

Definition at line 144 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnExposeCursor().

Referenced by clGUILofar::ConnectSignals().

8.65.1.11 gint WrapOnMotionLofar (GtkWidget * *gwSender*, GdkEventMotion * *gemEvent*, gpointer *gpData*)

Definition at line 151 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnMotionLofar().

Referenced by clGUILofar::ConnectSignals().

8.65.1.12 gint WrapOnPaletteActivate (GtkWidget * *gwSender*, gpointer *gpData*)

Definition at line 158 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnPaletteActivate().

8.65.1.13 void WrapOnAverageToggled (GtkToggleButton * *gtbSender*, gpointer *gpData*)

Definition at line 164 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnAverageToggled().

Referenced by clGUILofar::ConnectSignals().

8.65.1.14 void WrapOnClipValueChanged (GtkAdjustment * *gaSender*, gpointer *gpData*)

Definition at line 170 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnClipValueChanged().

Referenced by clGUILofar::ConnectSignals().

8.65.1.15 void WrapOnSaveClicks (GtkWidget * *gwSender*, gpointer *gpData*)

Definition at line 176 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnSaveClicks().

8.65.1.16 void WrapOnGdkInput (gpointer *gpData*, gint *giSource*, GdkInputCondition *gicCondition*)

Definition at line 182 of file GUILofar.cc.

References GUILofar, and clGUILofar::OnGdkInput().

8.65.2 Variable Documentation

8.65.2.1 const char* cpWindowTxt = "LOFAR / DEMON" `[static]`

Definition at line 38 of file GUILofar.cc.

Referenced by clGUILofar::Build(), and clGUILofar::clGUILofar().

8.65.2.2 const char* cpLServerTxt = "Server" `[static]`

Definition at line 40 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable1().

8.65.2.3 const char* cpaLChannelTxt[] = { "Channel", "Direction" } `[static]`

Definition at line 41 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable1().

8.65.2.4 const char* cpBConnectTxt = "Connect" `[static]`

Definition at line 42 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable1().

8.65.2.5 const char* cpBDisconnectTxt = "Disconnect" [static]

Definition at line 43 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable1().

8.65.2.6 const char* cpCBFreezeTxt = "Freeze" [static]

Definition at line 44 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable1().

8.65.2.7 const char* cpLTypeTxt = "Type" [static]

Definition at line 46 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.8 const char* cpaLTypeMenu[] = { "FFT", "Cepstrum", "Autocorr." } [static]

Definition at line 47 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.9 const char* cpLWindowTxt = "Window" [static]

Definition at line 48 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.10 const char* cpaLWindowMenu[] [static]

Initial value:

```
{ "Rectangle", "Bartlett",
   "Blackman", "Blackman-Harris", "Cosine tapered", "Exponential",
   "Flat-top", "Hamming", "Hanning", "Kaiser", "Kaiser-Bessel", "Tukey" }
```

Definition at line 49 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.11 const char* cpLWinParamTxt = "Window param." [static]

Definition at line 52 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.12 const char* cpLWinLengthTxt = "Window length" [static]

Definition at line 53 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.13 const char* cpaLWinLengthMenu[] [static]

Initial value:

```
{ "1024", "2048", "4096", "8192",
   "16384", "32768", "65536" }
```

Definition at line 54 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.14 const char* cpLLowerFreqTxt = "Lower freq." [static]

Definition at line 56 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.15 const char* cpLHigherFreqTxt = "Higher freq." [static]

Definition at line 57 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.16 const char* cpLOverlapTxt = "Overlap %" [static]

Definition at line 58 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable2().

8.65.2.17 const char* cpLRemoveNoiseTxt = "Remove noise" [static]

Definition at line 60 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.18 const char* cpaLRemoveNoiseMenu[] [static]

Initial value:

```
{ "None", "TPSW", "OTA",
   "Diff", "InvDiff", "StdDev" }
```

Definition at line 61 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.19 const char∗ cpLAlphaTxt = "Alpha" [static]

Definition at line 63 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.20 const char∗ cpLMeanLengthTxt = "Mean length" [static]

Definition at line 64 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.21 const char∗ cpLGapLengthTxt = "Gap length" [static]

Definition at line 65 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.22 const char∗ cpLAverageCountTxt = "Average count" [static]

Definition at line 66 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.23 const char∗ cpLClipTxt = "Clip" [static]

Definition at line 67 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.24 const char∗ cpaLClipMenu[] [static]

Initial value:

```
{ "None", "Low", "Both", "Mean" ,
   "Median", "10 dB", "20 dB", "50%", "75%", "Offset", "Offset 2",
   "Offset 3", "Sliding" }
```

Definition at line 68 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.25 const char∗ cpCBLinearTxt = "Linear" [static]

Definition at line 71 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.26 const char∗ cpCBDemonTxt = "DEMON" [static]

Definition at line 72 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.27 const char∗ cpLPaletteTxt = "Palette" [static]

Definition at line 73 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.28 const char∗ cpaLPaletteMenu[] [static]

Initial value:

```
{ "BW", "HSV", "Light", "Temp",
    "Dir", "Green", "Green2", "Green3", "Green4", "PureGreen", "WB" }
```

Definition at line 74 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.29 const char∗ cpCBAverageTxt = "ContAvg" [static]

Definition at line 76 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.30 const char∗ cpCBSavingTxt = "Saving" [static]

Definition at line 77 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.31 const char∗ cpBSaveTxt = "Save" [static]

Definition at line 78 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.32 const char* cpFSSaveTxt = "Save to TIFF file" [static]

Definition at line 79 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable3().

8.65.2.33 const char* cpLClipValueTxt = "Clip value" [static]

Definition at line 81 of file GUILofar.cc.

Referenced by clGUILofar::BuildTable4().

8.65.2.34 clGUILofar* GUILofar

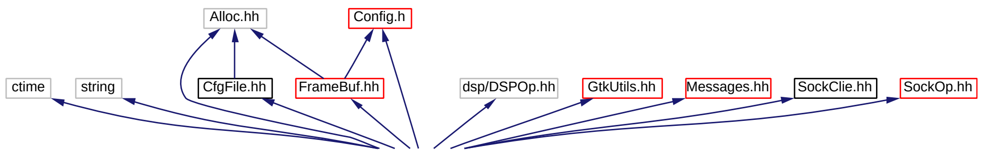
Definition at line 83 of file GUILofar.cc.

Referenced by main(), WrapOnAverageToggled(), WrapOnClipValueChanged(), WrapOnConfigureLofar(), WrapOnConnectClick(), WrapOnDelete(), WrapOnExposeCursor(), WrapOnExposeLine(), WrapOnExposeLofar(), WrapOnFreezeToggled(), WrapOnGdkInput(), WrapOnHideToggled(), WrapOnMotionLofar(), WrapOnPaletteActivate(), and WrapOnSaveClicks().

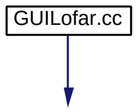
8.66 GUILofar.hh File Reference

```
#include <ctime>
#include <string>
#include <Alloc.hh>
#include <dsp/DSPOp.hh>
#include "Config.h"
#include "CfgFile.hh"
#include "FrameBuf.hh"
#include "GtkUtils.hh"
#include "Messages.hh"
#include "SockClie.hh"
#include "SockOp.hh"
```

Include dependency graph for GUILofar.hh:



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



Compounds

- struct **_stLofarCursor**

  Lofar cursor.

- class **clGUILofar**

  LOFAR/DEMON GUI.

Defines

- #define **LGUI_VER_MAJ** GLOBAL_VERSMAJ
- #define **LGUI_VER_MIN** GLOBAL_VERSMIN
- #define **LGUI_VER_PL** GLOBAL_VERSPL
- #define **LGUI_PADDING** 8
- #define **LGUI_ENTRY_WIDTH** 80
- #define **LGUI_LINESPECT_HEIGHT** 40
- #define **LGUI_CURSOR_HEIGHT** 16
- #define **LGUI_SERVER_MAXLEN** 255
- #define **LGUI_CH_LOWER** 1
- #define **LGUI_CH_UPPER** 16
- #define **LGUI_LOFAR_BG** 0x00ffffff
- #define **LGUI_LOFAR_FG** 0x00000000
- #define **LGUI_LINE_BG** 0x00ffffff
- #define **LGUI_LINE_FG** 0x000000ff
- #define **LGUI_CURSOR_BG** 0x00ffffff
- #define **LGUI_CURSOR_FG** 0x00ff0000
- #define **LGUI_CONV_BUF_SIZE** 255
- #define **LGUI_TYPE_ITEMS** 3
- #define **LGUI_WINDOW_ITEMS** 12
- #define **LGUI_WIN_LENGTH_ITEMS** 7
- #define **LGUI_REMOVE_NOISE_ITEMS** 6
- #define **LGUI_CLIP_ITEMS** 13
- #define **LGUI_PALETTE_ITEMS** 11

Typedefs

- typedef **_stLofarCursor stLofarCursor**

  Lofar cursor.

- typedef **_stLofarCursor** ∗ **stpLofarCursor**

  Lofar cursor.

Enumerations

- enum {

  LGUI_PAL_BW = 0, **LGUI_PAL_HSV** = 1, **LGUI_PAL_LIGHT** = 2, **LGUI_PAL_TEMP** = 3,

  LGUI_PAL_DIR = 4, **LGUI_PAL_GREEN** = 5, **LGUI_PAL_GREEN2** = 6, **LGUI_PAL_GREEN3** = 7,

  LGUI_PAL_GREEN4 = 8, **LGUI_PAL_PUREGREEN** = 9, **LGUI_PAL_WB** = 10 }

  Palettes available.

- enum { **LGUI_FIT_NONE** = 0, **LGUI_FIT_NEIGHBOR** = 1, **LGUI_FIT_AVERAGE** = 2 }

  Fit algorithms available.

- enum { **LGUI_CURSOR_11** = 0, **LGUI_CURSOR_INF** = 1 }

  Cursor types.

8.66.1 Define Documentation

8.66.1.1 #define LGUI_VER_MAJ GLOBAL_VERSMAJ

Definition at line 41 of file GUILofar.hh.

Referenced by clGUILofar::clGUILofar().

8.66.1.2 #define LGUI_VER_MIN GLOBAL_VERSMIN

Definition at line 42 of file GUILofar.hh.

Referenced by clGUILofar::clGUILofar().

8.66.1.3 #define LGUI_VER_PL GLOBAL_VERSPL

Definition at line 43 of file GUILofar.hh.

Referenced by clGUILofar::clGUILofar().

8.66.1.4 #define LGUI_PADDING 8

Definition at line 44 of file GUILofar.hh.

Referenced by clGUILofar::Build(), clGUILofar::BuildTable1(), clGUILofar::BuildTable2(), clGUILofar::BuildTable3(), and clGUILofar::BuildTable4().

8.66.1.5 #define LGUI_ENTRY_WIDTH 80

Definition at line 45 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable2(), and clGUILofar::BuildTable3().

8.66.1.6 #define LGUI_LINESPECT_HEIGHT 40

Definition at line 46 of file GUILofar.hh.

Referenced by clGUILofar::BuildTableLofar().

8.66.1.7 #define LGUI_CURSOR_HEIGHT 16

Definition at line 47 of file GUILofar.hh.

Referenced by clGUILofar::BuildTableLofar().

8.66.1.8 #define LGUI_SERVER_MAXLEN 255

Definition at line 48 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable1(), clGUILofar::OnConnectClick(), and clGUILofar::ParseServerStr().

8.66.1.9 #define LGUI_CH_LOWER 1

Definition at line 49 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable1().

8.66.1.10 #define LGUI_CH_UPPER 16

Definition at line 50 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable1().

8.66.1.11 #define LGUI_LOFAR_BG 0x00ffffff

Definition at line 51 of file GUILofar.hh.

Referenced by clGUILofar::BuildDrawingPrims().

8.66.1.12 #define LGUI_LOFAR_FG 0x00000000

Definition at line 52 of file GUILofar.hh.

Referenced by clGUILofar::BuildDrawingPrims().

8.66.1.13 #define LGUI_LINE_BG 0x00ffffff

Definition at line 53 of file GUILofar.hh.

Referenced by clGUILofar::BuildDrawingPrims().

8.66.1.14 #define LGUI_LINE_FG 0x000000ff

Definition at line 54 of file GUILofar.hh.

Referenced by clGUILofar::BuildDrawingPrims().

8.66.1.15 #define LGUI_CURSOR_BG 0x00ffffff

Definition at line 55 of file GUILofar.hh.

Referenced by clGUILofar::BuildDrawingPrims().

8.66.1.16 #define LGUI_CURSOR_FG 0x00ff0000

Definition at line 56 of file GUILofar.hh.

Referenced by clGUILofar::BuildDrawingPrims().

8.66.1.17 #define LGUI_CONV_BUF_SIZE 255

Definition at line 57 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable2(), clGUILofar::BuildTable3(), clGUILofar::Configure(), clGUILofar::OnGdkInput(), and clGUILofar::PrintStatus().

8.66.1.18 #define LGUI_TYPE_ITEMS 3

Definition at line 59 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable2(), and clGUILofar::SendSettings().

8.66.1.19 #define LGUI_WINDOW_ITEMS 12

Definition at line 60 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable2(), and clGUILofar::SendSettings().

8.66.1.20 #define LGUI_WIN_LENGTH_ITEMS 7

Definition at line 61 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable2(), and clGUILofar::SendSettings().

8.66.1.21 #define LGUI_REMOVE_NOISE_ITEMS 6

Definition at line 62 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable3(), and clGUILofar::SendSettings().

8.66.1.22 #define LGUI_CLIP_ITEMS 13

Definition at line 63 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable3(), and clGUILofar::SendSettings().

8.66.1.23 #define LGUI_PALETTE_ITEMS 11

Definition at line 64 of file GUILofar.hh.

Referenced by clGUILofar::BuildTable3(), clGUILofar::ConnectSignals(), and clGUILofar::OnPaletteActivate().

8.66.2 Typedef Documentation

8.66.2.1 typedef struct _stLofarCursor stLofarCursor

Lofar cursor.

8.66.2.2 typedef struct _stLofarCursor * stpLofarCursor

Lofar cursor.

8.66.3 Enumeration Type Documentation

8.66.3.1 anonymous enum

Palettes available.

Enumeration values:

- **LGUI_PAL_BW**
- **LGUI_PAL_HSV**
- **LGUI_PAL_LIGHT**
- **LGUI_PAL_TEMP**
- **LGUI_PAL_DIR**
- **LGUI_PAL_GREEN**
- **LGUI_PAL_GREEN2**
- **LGUI_PAL_GREEN3**
- **LGUI_PAL_GREEN4**
- **LGUI_PAL_PUREGREEN**
- **LGUI_PAL_WB**

Definition at line 70 of file GUILofar.hh.

8.66.3.2 anonymous enum

Fit algorithms available.

Enumeration values:

- **LGUI_FIT_NONE**

LGUI_FIT_NEIGHBOR

LGUI_FIT_AVERAGE

Definition at line 88 of file GUILofar.hh.

8.66.3.3 anonymous enum

Cursor types.

Enumeration values:

LGUI_CURSOR_11 11 spike type

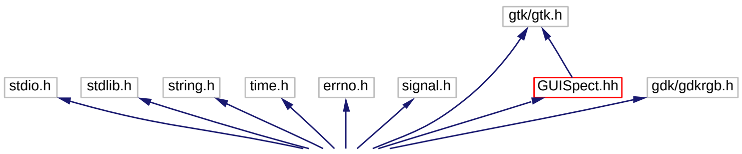
LGUI_CURSOR_INF Inifinite spike count.

Definition at line 98 of file GUILofar.hh.

8.67 GUISpect.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <errno.h>
#include <signal.h>
#include <gtk/gtk.h>
#include <gdk/gdkrgb.h>
#include "GUISpect.hh"
```

Include dependency graph for GUISpect.cc:



Functions

- **G_LOCK_DEFINE_STATIC** (gmInputMutex)
- int **main** (int argc, char *argv[])
- gint **WrapOnDelete** (GtkWidget *gwSender, GdkEventAny *geaEvent)
- void **WrapOnHideToggled** (GtkToggleButton *gtbSender, gpointer *gpData)
- gint **WrapOnConnectClick** (GtkWidget *gwSender, gpointer gpData)
- void **WrapOnFreezeToggled** (GtkToggleButton *gtbSender, gpointer *gpData)
- gint **WrapOnPaletteActivate** (GtkWidget *gwSender, gpointer gpData)
- gint **WrapOnMotionSgram** (GtkWidget *gwSender, GdkEventMotion *gemEvent)
- gint **WrapOnMotionSpect** (GtkWidget *gwSender, GdkEventMotion *gemEvent)
- gint **WrapOnExposeSgram** (GtkWidget *gwSender, GdkEventExpose *geeEvent)
- gint **WrapOnExposeSpect** (GtkWidget *gwSender, GdkEventExpose *geeEvent)
- gboolean **WrapOnConfigure** (GtkWidget *gwSender, GdkEventConfigure *gecEvent, gpointer gpData)

- void **WrapOnSizeAllocate** (GtkWidget ∗gwSender, GtkAllocation ∗gaAllocation, gpointer gpData)
- void **WrapGdkInput** (gpointer gpData, gint giSource, GdkInputCondition gicCondition)
- void **WrapOnSaveClicks** (GtkButton ∗gbSender, gpointer gpData)

Variables

- const char ∗ **cpWindowTxt** = "Spectrogram"
- const char ∗ **cpLServerTxt** = "Server"
- const char ∗ **cpaLChannelTxt** [ ] = { "Channel", "Direction" }
- const char ∗ **cpBConnectTxt** = "Connect"
- const char ∗ **cpBDisconnectTxt** = "Disconnect"
- const char ∗ **cpCBFreezeTxt** = "Freeze"
- const char ∗ **cpLTypeTxt** = "Type"
- const char ∗ **cpaLTypeMenu** [ ]
- const char ∗ **cpLWindowTxt** = "Window"
- const char ∗ **cpaLWindowMenu** [ ]
- const char ∗ **cpLWindowParamTxt** = "Window param."
- const char ∗ **cpLWindowLenTxt** = "Window length"
- const char ∗ **cpaLWindowLenMenu** [ ]
- const char ∗ **cpLLowFreqTxt** = "Lower freq."
- const char ∗ **cpLHighFreqTxt** = "Higher freq."
- const char ∗ **cpLGainTxt** = "Gain dB"
- const char ∗ **cpLSlopeTxt** = "Gain dB/oct"
- const char ∗ **cpLOverlapTxt** = "Overlap (%)"
- const char ∗ **cpLLinearTxt** = "Linear scale"
- const char ∗ **cpLNormalizeTxt** = "Normalize"
- const char ∗ **cpLRemoveNoiseTxt** = "Remove noise"
- const char ∗ **cpaLRemoveNoiseMenu** [ ]
- const char ∗ **cpLAlphaTxt** = "Alpha"
- const char ∗ **cpLMeanLengthTxt** = "Mean length"
- const char ∗ **cpLGapLengthTxt** = "Gap length"
- const char ∗ **cpLDynRangeTxt** = "Dynamic range (dB)"
- const char ∗ **cpLPaletteTxt** = "Palette"
- const char ∗ **cpaLPaletteMenu** [ ]
- const char ∗ **cpBSaveTxt** = "Save"
- const char ∗ **cpFSSaveTxt** = "Save to TIFF file"
- **clSpectGUI ∗ SpectGUI**

8.67.1 Function Documentation

8.67.1.1 G_LOCK_DEFINE_STATIC (gmInputMutex)

8.67.1.2 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 84 of file GUISpect.cc.

References clSpectGUI::Exec(), and SpectGUI.

8.67.1.3 gint WrapOnDelete (GtkWidget * *gwSender*, GdkEventAny * *geaEvent*)

Definition at line 97 of file GUISpect.cc.

References clSpectGUI::OnDelete(), and SpectGUI.

8.67.1.4 void WrapOnHideToggled (GtkToggleButton * *gtbSender*, gpointer * *gpData*)

Definition at line 103 of file GUISpect.cc.

References clSpectGUI::OnHideToggled(), and SpectGUI.

8.67.1.5 gint WrapOnConnectClick (GtkWidget * *gwSender*, gpointer *gpData*)

Definition at line 109 of file GUISpect.cc.

References clSpectGUI::OnConnectClick(), and SpectGUI.

8.67.1.6 void WrapOnFreezeToggled (GtkToggleButton * *gtbSender*, gpointer * *gpData*)

Definition at line 115 of file GUISpect.cc.

References clSpectGUI::OnFreezeToggled(), and SpectGUI.

8.67.1.7 gint WrapOnPaletteActivate (GtkWidget * *gwSender*, gpointer *gpData*)

Definition at line 121 of file GUISpect.cc.

References clSpectGUI::OnPaletteActivate(), and SpectGUI.

8.67.1.8 gint WrapOnMotionSgram (GtkWidget * *gwSender*, GdkEventMotion * *gemEvent*)

Definition at line 127 of file GUISpect.cc.

References clSpectGUI::OnMotionSgram(), and SpectGUI.

Referenced by clSpectGUI::ConnectSignals().

8.67.1.9 gint WrapOnMotionSpect (GtkWidget * *gwSender*, GdkEventMotion * *gemEvent*)

Definition at line 133 of file GUISpect.cc.

References clSpectGUI::OnMotionSpect(), and SpectGUI.

Referenced by clSpectGUI::ConnectSignals().

8.67.1.10 gint WrapOnExposeSgram (GtkWidget * *gwSender*, GdkEventExpose * *geeEvent*)

Definition at line 139 of file GUISpect.cc.

References clSpectGUI::OnExposeSgram(), and SpectGUI.

Referenced by clSpectGUI::ConnectSignals().

8.67.1.11 gint WrapOnExposeSpect (GtkWidget * *gwSender*, GdkEventExpose * *geeEvent*)

Definition at line 145 of file GUISpect.cc.

References clSpectGUI::OnExposeSpect(), and SpectGUI.

Referenced by clSpectGUI::ConnectSignals().

8.67.1.12 gboolean WrapOnConfigure (GtkWidget * *gwSender*, GdkEventConfigure * *gecEvent*, gpointer *gpData*)

Definition at line 151 of file GUISpect.cc.

References clSpectGUI::OnConfigure(), and SpectGUI.

8.67.1.13 void WrapOnSizeAllocate (GtkWidget * *gwSender*, GtkAllocation * *gaAllocation*, gpointer *gpData*)

Definition at line 158 of file GUISpect.cc.

References clSpectGUI::OnSizeAllocate(), and SpectGUI.

Referenced by clSpectGUI::ConnectSignals().

8.67.1.14 void WrapGdkInput (gpointer *gpData*, gint *giSource*, GdkInputCondition *gicCondition*)

Definition at line 165 of file GUISpect.cc.

References clSpectGUI::GdkInput(), and SpectGUI.

Referenced by clSpectGUI::InitConnection().

8.67.1.15 void WrapOnSaveClicks (GtkButton * *gbSender*, gpointer *gpData*)

Definition at line 172 of file GUISpect.cc.

References clSpectGUI::OnSaveClicks(), and SpectGUI.

8.67.2 Variable Documentation

8.67.2.1 const char* cpWindowTxt = "Spectrogram" [static]

Definition at line 42 of file GUISpect.cc.

Referenced by clSpectGUI::Build(), and clSpectGUI::clSpectGUI().

8.67.2.2 const char* cpLServerTxt = "Server" [static]

Definition at line 43 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable1().

8.67.2.3 const char* cpaLChannelTxt[] = { "Channel", "Direction" } [static]

Definition at line 44 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable1().

8.67.2.4 const char* cpBConnectTxt = "Connect" [static]

Definition at line 45 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable1().

8.67.2.5 const char* cpBDisconnectTxt = "Disconnect" [static]

Definition at line 46 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable1().

8.67.2.6 const char* cpCBFreezeTxt = "Freeze" [static]

Definition at line 47 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable1().

8.67.2.7 const char* cpLTypeTxt = "Type" [static]

Definition at line 48 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.8 const char* cpaLTypeMenu[] [static]

Initial value:

```
{ "STFT", "(MR-STFT)", "(Gabor)",
   "(WVD)", "Hankel", "Autocorr.", "Cepstrum" }
```

Definition at line 49 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.9 const char* cpLWindowTxt = "Window" [static]

Definition at line 51 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.10 const char* cpaLWindowMenu[] [static]

Initial value:

```
{ "Rectangle", "Bartlett",
   "Blackman", "Blackman-Harris", "Cosine tapered", "Exponential",
   "Flat top", "Generic cosine", "Hamming", "Hanning", "Kaiser",
   "Kaiser-Bessel", "Tukey" }
```

Definition at line 52 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.11 const char* cpLWindowParamTxt = "Window param." [static]

Definition at line 56 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.12 const char* cpLWindowLenTxt = "Window length" [static]

Definition at line 57 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.13 const char* cpaLWindowLenMenu[] [static]

Initial value:

```
{ "128", "256", "512", "1024",
   "2048", "4096", "8192", "16384", "32768", "65536" }
```

Definition at line 58 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2(), and clSpectGUI::GetSettings().

8.67.2.14 const char* cpLLowFreqTxt = "Lower freq." [static]

Definition at line 60 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.15 const char* cpLHighFreqTxt = "Higher freq." [static]

Definition at line 61 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.16 const char* cpLGainTxt = "Gain dB" [static]

Definition at line 62 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.17 const char* cpLSlopeTxt = "Gain dB/oct" [static]

Definition at line 63 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.18 const char* cpLOverlapTxt = "Overlap (%)" [static]

Definition at line 64 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.19 const char* cpLLinearTxt = "Linear scale" [static]

Definition at line 65 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.20 const char* cpLNormalizeTxt = "Normalize" [static]

Definition at line 66 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable2().

8.67.2.21 const char* cpLRemoveNoiseTxt = "Remove noise" [static]

Definition at line 68 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.22 const char* cpaLRemoveNoiseMenu[] [static]

Initial value:

```
{ "None", "TPSW", "OTA",
   "Diff", "InvDiff" }
```

Definition at line 69 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.23 const char* cpLAlphaTxt = "Alpha" [static]

Definition at line 71 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.24 const char* cpLMeanLengthTxt = "Mean length" [static]

Definition at line 72 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.25 const char* cpLGapLengthTxt = "Gap length" [static]

Definition at line 73 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.26 const char* cpLDynRangeTxt = "Dynamic range (dB)" [static]

Definition at line 74 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.27 const char* cpLPaletteTxt = "Palette" [static]

Definition at line 75 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.28 const char* cpaLPaletteMenu[] [static]

Initial value:

```
{ "BW", "HSV", "Light", "Temp",
   "Dir", "Green", "Green2", "PureGreen", "WB" }
```

Definition at line 76 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.29 const char* cpBSaveTxt = "Save" [static]

Definition at line 78 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.30 const char* cpFSSaveTxt = "Save to TIFF file" [static]

Definition at line 79 of file GUISpect.cc.

Referenced by clSpectGUI::BuildTable3().

8.67.2.31 clSpectGUI* SpectGUI

Definition at line 81 of file GUISpect.cc.

Referenced by main(), WrapGdkInput(), WrapOnConfigure(), WrapOnConnectClick(), WrapOnDelete(), WrapOnExposeSgram(), WrapOnExposeSpect(), WrapOnFreezeToggled(), WrapOnHideToggled(), WrapOnMotionSgram(), WrapOnMotionSpect(), WrapOnPaletteActivate(), WrapOnSaveClicks(), and WrapOnSizeAllocate().

8.68 GUISpect.hh File Reference

`#include <gtk/gtk.h>`

`#include <Alloc.hh>`

`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

`#include "CfgFile.hh"`

`#include "GtkUtils.hh"`

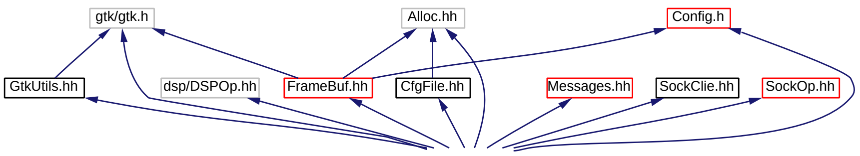
`#include "Messages.hh"`

`#include "SockClie.hh"`

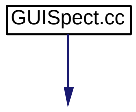
`#include "SockOp.hh"`

`#include "FrameBuf.hh"`

Include dependency graph for GUISpect.hh:



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



Compounds

- class **clSpectGUI**

    Transient spectrum GUI.

Defines

- #define **SGUI_VER_MAJ** GLOBAL_VERSMAJ
- #define **SGUI_VER_MIN** GLOBAL_VERSMIN
- #define **SGUI_VER_PL** GLOBAL_VERSPL

- #define **SGUI_CONV_MAXLEN** 255
- #define **SGUI_DEF_WIDTH** 1024
- #define **SGUI_DEF_HEIGHT** 700
- #define **SGUI_PADDING** 8
- #define **SGUI_SERVER_MAXLEN** 255
- #define **SGUI_CH_LOWER** 1.0
- #define **SGUI_CH_UPPER** 16.0
- #define **SGUI_GRAM_BG** 0x00ffffff
- #define **SGUI_GRAM_FG** 0x00000000
- #define **SGUI_SPECT_BG** 0x00ffffff
- #define **SGUI_SPECT_FG** 0x000000ff
- #define **SGUI_TYPE_ITEMS** 7
- #define **SGUI_WINDOW_ITEMS** 13
- #define **SGUI_WINLEN_ITEMS** 10
- #define **SGUI_REMOVE_NOISE_ITEMS** 5
- #define **SGUI_PALETTE_ITEMS** 9

Enumerations

- enum { **SGUI_FIT_NONE** = 0, **SGUI_FIT_NEIGHBOR** = 1, **SGUI_FIT_AVERAGE** = 2 }

  Fit types available.

- enum {

  SGUI_PAL_BW = 0, **SGUI_PAL_HSV** = 1, **SGUI_PAL_LIGHT** = 2, **SGUI_PAL_TEMP** = 3,

  SGUI_PAL_DIR = 4, **SGUI_PAL_GREEN** = 5, **SGUI_PAL_GREEN2** = 6, **SGUI_PAL_PUREGREEN** = 7,

  SGUI_PAL_WB = 8 }

  Palettes available.

8.68.1 Define Documentation

8.68.1.1 #define SGUI_VER_MAJ GLOBAL_VERSMAJ

Definition at line 41 of file GUISpect.hh.

Referenced by clSpectGUI::clSpectGUI().

8.68.1.2 #define SGUI_VER_MIN GLOBAL_VERSMIN

Definition at line 42 of file GUISpect.hh.

Referenced by clSpectGUI::clSpectGUI().

8.68.1.3 #define SGUI_VER_PL GLOBAL_VERSPL

Definition at line 43 of file GUISpect.hh.

Referenced by clSpectGUI::clSpectGUI().

8.68.1.4 #define SGUI_CONV_MAXLEN 255

Definition at line 44 of file GUISpect.hh.

Referenced by clSpectGUI::Build(), clSpectGUI::GdkInput(), clSpectGUI::OnMotionSgram(), and clSpectGUI::OnMotionSpect().

8.68.1.5 #define SGUI_DEF_WIDTH 1024

Definition at line 45 of file GUISpect.hh.

Referenced by clSpectGUI::Build().

8.68.1.6 #define SGUI_DEF_HEIGHT 700

Definition at line 46 of file GUISpect.hh.

Referenced by clSpectGUI::Build().

8.68.1.7 #define SGUI_PADDING 8

Definition at line 47 of file GUISpect.hh.

Referenced by clSpectGUI::Build(), clSpectGUI::BuildTable1(), clSpectGUI::BuildTable2(), and clSpectGUI::BuildTable3().

8.68.1.8 #define SGUI_SERVER_MAXLEN 255

Definition at line 48 of file GUISpect.hh.

Referenced by clSpectGUI::BuildTable1(), clSpectGUI::OnConnectClick(), and clSpectGUI::ParseServerStr().

8.68.1.9 #define SGUI_CH_LOWER 1.0

Definition at line 49 of file GUISpect.hh.

Referenced by clSpectGUI::BuildTable1().

8.68.1.10 #define SGUI_CH_UPPER 16.0

Definition at line 50 of file GUISpect.hh.

Referenced by clSpectGUI::BuildTable1().

8.68.1.11 #define SGUI_GRAM_BG 0x00ffffff

Definition at line 51 of file GUISpect.hh.

Referenced by clSpectGUI::BuildDrawingPrims().

8.68.1.12 #define SGUI_GRAM_FG 0x00000000

Definition at line 52 of file GUISpect.hh.

Referenced by clSpectGUI::BuildDrawingPrims().

8.68.1.13 #define SGUI_SPECT_BG 0x00ffffff

Definition at line 53 of file GUISpect.hh.

Referenced by clSpectGUI::BuildDrawingPrims().

8.68.1.14 #define SGUI_SPECT_FG 0x000000ff

Definition at line 54 of file GUISpect.hh.

Referenced by clSpectGUI::BuildDrawingPrims().

8.68.1.15 #define SGUI_TYPE_ITEMS 7

Definition at line 56 of file GUISpect.hh.

Referenced by clSpectGUI::BuildTable2(), and clSpectGUI::GetSettings().

8.68.1.16 #define SGUI_WINDOW_ITEMS 13

Definition at line 57 of file GUISpect.hh.

Referenced by clSpectGUI::BuildTable2(), and clSpectGUI::GetSettings().

8.68.1.17 #define SGUI_WINLEN_ITEMS 10

Definition at line 58 of file GUISpect.hh.

Referenced by clSpectGUI::BuildTable2(), and clSpectGUI::GetSettings().

8.68.1.18 #define SGUI_REMOVE_NOISE_ITEMS 5

Definition at line 59 of file GUISpect.hh.

Referenced by clSpectGUI::BuildTable3(), and clSpectGUI::GetSettings().

8.68.1.19 #define SGUI_PALETTE_ITEMS 9

Definition at line 60 of file GUISpect.hh.

Referenced by clSpectGUI::BuildTable3(), and clSpectGUI::ConnectSignals().

8.68.2 Enumeration Type Documentation

8.68.2.1 anonymous enum

Fit types available.

Enumeration values:

SGUI_FIT_NONE

SGUI_FIT_NEIGHBOR

SGUI_FIT_AVERAGE

Definition at line 66 of file GUISpect.hh.

8.68.2.2 anonymous enum

Palettes available.

Enumeration values:

SGUI_PAL_BW

SGUI_PAL_HSV

SGUI_PAL_LIGHT

SGUI_PAL_TEMP

SGUI_PAL_DIR

SGUI_PAL_GREEN

SGUI_PAL_GREEN2

SGUI_PAL_PUREGREEN

SGUI_PAL_WB

Definition at line 77 of file GUISpect.hh.

8.69 GUITrans.cc File Reference

`#include <stdlib.h>`

`#include <string.h>`

`#include <signal.h>`

`#include <unistd.h>`

`#include <math.h>`

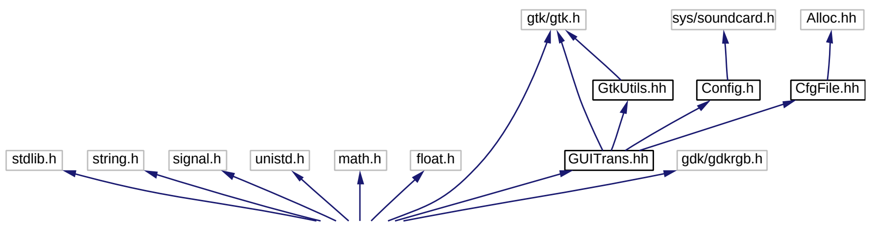
`#include <float.h>`

`#include <gtk/gtk.h>`

`#include <gdk/gdkrgb.h>`

`#include "GUITrans.hh"`

Include dependency graph for GUITrans.cc:



Functions

- int **main** (int argc, char ∗argv[ ])
- gboolean **WrapOnDelete** (GtkWidget ∗gwSender, GdkEvent ∗geEvent, gpointer gpData)
- void **WrapOnConnectClick** (GtkButton ∗gbSender, gpointer gpData)

Variables

- const char ∗ **cpWindowTxt** = "Locate"
- const char ∗ **cpLServerTxt** = "Server"
- const char ∗ **cpBConnectTxt** = "Connect"
- **clGUITransient GUITransient**

8.69.1 Function Documentation

8.69.1.1 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 45 of file GUITrans.cc.

References GUITransient, and clGUITransient::Main().

8.69.1.2 gboolean WrapOnDelete (GtkWidget * *gwSender*, GdkEvent * *geEvent*, gpointer *gpData*)

Definition at line 53 of file GUITrans.cc.

References GUITransient, and clGUITransient::OnDelete().

8.69.1.3 void WrapOnConnectClick (GtkButton * *gbSender*, gpointer *gpData*)

Definition at line 60 of file GUITrans.cc.

References GUITransient, and clGUITransient::OnConnectClick().

8.69.2 Variable Documentation

8.69.2.1 const char* cpWindowTxt = "Locate" [static]

Definition at line 36 of file GUITrans.cc.

Referenced by clGUITransient::Build(), and clGUITransient::Main().

8.69.2.2 const char* cpLServerTxt = "Server" [static]

Definition at line 38 of file GUITrans.cc.

Referenced by clGUITransient::BuildTable1().

8.69.2.3 const char* cpBConnectTxt = "Connect" [static]

Definition at line 39 of file GUITrans.cc.

Referenced by clGUITransient::BuildTable1().

8.69.2.4 clGUITransient GUITransient

Definition at line 42 of file GUITrans.cc.

Referenced by main(), WrapOnConnectClick(), and WrapOnDelete().

8.70 GUITrans.hh File Reference

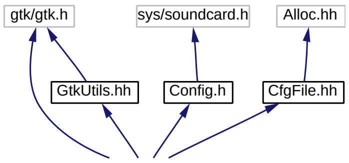
`#include <gtk/gtk.h>`

`#include "Config.h"`

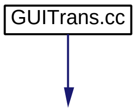
`#include "CfgFile.hh"`

`#include "GtkUtils.hh"`

Include dependency graph for GUITrans.hh:



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



Compounds

- class **clGUITransient**

Defines

- #define **GUITRANS_VER_MAJ** GLOBAL_VERSMAJ
- #define **GUITRANS_VER_MIN** GLOBAL_VERSMIN
- #define **GUITRANS_VER_PL** GLOBAL_VERSPL
- #define **GUITRANS_WSPACING** 8
- #define **GUITRANS_ENTRY_WIDTH** 80
- #define **GUITRANS_SERVER_MAXLEN** 256
- #define **GUITRANS_LOCATE_BG** 0x00000000
- #define **GUITRANS_LOCATE_FG** 0x00ffffff
- #define **GUITRANS_PALETTE_ITEMS** 9

Enumerations

- enum {
  GUITRANS_PAL_BW = 0, **GUITRANS_PAL_HSV** = 1, **GUITRANS_PAL_LIGHT** = 2, **GUITRANS_PAL_TEMP** = 3,
  GUITRANS_PAL_DIR = 4, **GUITRANS_PAL_GREEN** = 5, **GUITRANS_PAL_GREEN2** = 6, **GUITRANS_PAL_PUREGREEN** = 7,
  GUITRANS_PAL_WB = 8 }

  Available palettes.

8.70.1 Define Documentation

8.70.1.1 #define GUITRANS_VER_MAJ GLOBAL_VERSMAJ

Definition at line 33 of file GUITrans.hh.

Referenced by clGUITransient::Main().

8.70.1.2 #define GUITRANS_VER_MIN GLOBAL_VERSMIN

Definition at line 34 of file GUITrans.hh.

Referenced by clGUITransient::Main().

8.70.1.3 #define GUITRANS_VER_PL GLOBAL_VERSPL

Definition at line 35 of file GUITrans.hh.

Referenced by clGUITransient::Main().

8.70.1.4 #define GUITRANS_WSPACING 8

Definition at line 36 of file GUITrans.hh.

Referenced by clGUITransient::Build(), and clGUITransient::BuildTable1().

8.70.1.5 #define GUITRANS_ENTRY_WIDTH 80

Definition at line 37 of file GUITrans.hh.

8.70.1.6 #define GUITRANS_SERVER_MAXLEN 256

Definition at line 38 of file GUITrans.hh.

Referenced by clGUITransient::BuildTable1().

8.70.1.7 #define GUITRANS_LOCATE_BG 0x00000000

Definition at line 39 of file GUITrans.hh.

8.70.1.8 #define GUITRANS_LOCATE_FG 0x00ffffff

Definition at line 40 of file GUITrans.hh.

8.70.1.9 #define GUITRANS_PALETTE_ITEMS 9

Definition at line 42 of file GUITrans.hh.

8.70.2 Enumeration Type Documentation

8.70.2.1 anonymous enum

Available palettes.

Enumeration values:

- **GUITRANS_PAL_BW**
- **GUITRANS_PAL_HSV**
- **GUITRANS_PAL_LIGHT**
- **GUITRANS_PAL_TEMP**
- **GUITRANS_PAL_DIR**
- **GUITRANS_PAL_GREEN**
- **GUITRANS_PAL_GREEN2**
- **GUITRANS_PAL_PUREGREEN**
- **GUITRANS_PAL_WB**

Definition at line 48 of file GUITrans.hh.

8.71 Level.cc File Reference

`#include <stdio.h>`

`#include <string.h>`

`#include <signal.h>`

`#include <math.h>`

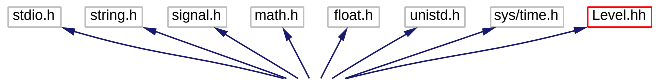
`#include <float.h>`

`#include <unistd.h>`

`#include <sys/time.h>`

`#include "Level.hh"`

Include dependency graph for Level.cc:



Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char ∗argv[ ])

Variables

- bool **bDebug** = false
- bool **bDaemon** = false
- **clLevel** ∗ **Level**

8.71.1 Function Documentation

8.71.1.1 void SigHandler (int *iSigNum*)

Definition at line 39 of file Level.cc.

References bDebug.

8.71.1.2 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 55 of file Level.cc.

References bDaemon, bDebug, Level, clLevel::Main(), and SigHandler().

8.71.2 Variable Documentation

8.71.2.1 bool bDebug = false `[static]`

Definition at line 34 of file Level.cc.

Referenced by clLevel::Initialize(), main(), and SigHandler().

8.71.2.2 bool bDaemon = false `[static]`

Definition at line 35 of file Level.cc.

Referenced by main().

8.71.2.3 clLevel* Level

Definition at line 36 of file Level.cc.

Referenced by main().

8.72 Level.hh File Reference

`#include <limits.h>`

`#include <Alloc.hh>`

`#include <dsp/DSPOp.hh>`

`#include <dsp/Filter.hh>`

`#include "Config.h"`

`#include "LocalMsg.h"`

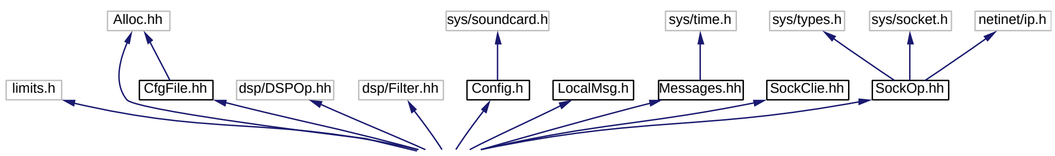
`#include "CfgFile.hh"`

`#include "Messages.hh"`

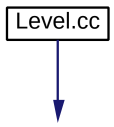
`#include "SockClie.hh"`

`#include "SockOp.hh"`

Include dependency graph for Level.hh:



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



Compounds

- class **clLevel**

  Level server.

Defines

- #define **LEVEL_FILTER_WINDOW** true

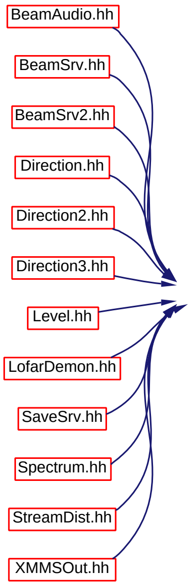
8.72.1 Define Documentation

8.72.1.1 #define LEVEL_FILTER_WINDOW true

Definition at line 40 of file Level.hh.

8.73 LocalMsg.h File Reference

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



Compounds

- struct **_stRawDataFirst**

  Header from streamdist server.

- struct **_stRawDataReq**

  Request to streamdist server.

Typedefs

- typedef **_stRawDataFirst stRawDataFirst**

  Header from streamdist server.

- typedef **_stRawDataFirst** $*$ **stpRawDataFirst**

  Header from streamdist server.

- typedef **_stRawDataReq stRawDataReq**

  Request to streamdist server.

- typedef **_stRawDataReq** $*$ **stpRawDataReq**

  Request to streamdist server.

8.73.1 Typedef Documentation

8.73.1.1 typedef struct _stRawDataFirst stRawDataFirst

Header from streamdist server.

8.73.1.2 typedef struct _stRawDataFirst * stpRawDataFirst

Header from streamdist server.

8.73.1.3 typedef struct _stRawDataReq stRawDataReq

Request to streamdist server.

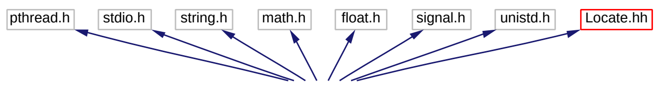
8.73.1.4 typedef struct _stRawDataReq * stpRawDataReq

Request to streamdist server.

8.74 Locate.cc File Reference

```
#include <pthread.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <float.h>
#include <signal.h>
#include <unistd.h>
#include "Locate.hh"
```

Include dependency graph for Locate.cc:



Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char ∗argv[ ])
- void **MasterSigHandler** (int iSigNum)
- void ∗ **WrapProcessThread** (void ∗vpParam)
- void ∗ **WrapServeClientThread** (void ∗vpParam)

Variables

- bool **bDebug** = false
- int **iNodeCount**
- int **iRank**
- **clMPIProc MPIProc**
- **clLocate** ∗ **Locate**
- **clSubLocate** ∗ **SubLocate**

8.74.1 Function Documentation

8.74.1.1 void SigHandler (int *iSigNum*)

Definition at line 46 of file Locate.cc.

References bDebug.

8.74.1.2 int main (int *argc*, char * *argv*[ ])

Definition at line 62 of file Locate.cc.

References bDebug, clMPIProc::Finalize(), clMPIProc::GetNodeCount(), clMPIProc::GetRank(), GLOBAL_VERSMAJ, GLOBAL_VERSMIN, GLOBAL_VERSPL, clMPIProc::Initialize(), iNodeCount, iRank, Locate, clSubLocate::Main(), clLocate::Main(), MPIProc, SigHandler(), and SubLocate.

8.74.1.3 void MasterSigHandler (int *iSigNum*)

Definition at line 110 of file Locate.cc.

References Locate, and clLocate::Stop().

Referenced by clLocate::Main().

8.74.1.4 void* WrapProcessThread (void * *vpParam*)

Definition at line 125 of file Locate.cc.

References Locate, and clLocate::ProcessThread().

Referenced by clLocate::Main().

8.74.1.5 void* WrapServeClientThread (void * *vpParam*)

Definition at line 131 of file Locate.cc.

References Locate, and clLocate::ServeClientThread().

8.74.2 Variable Documentation

8.74.2.1 bool bDebug = false `[static]`

Definition at line 38 of file Locate.cc.

Referenced by clSubLocate::Main(), clLocate::Main(), main(), clLocate::ProcessThread(), and SigHandler().

8.74.2.2 int iNodeCount `[static]`

Definition at line 39 of file Locate.cc.

Referenced by clLocate::Initialize(), main(), clLocate::ProcessThread(), and clLocate::Stop().

8.74.2.3 int iRank [static]

Definition at line 40 of file Locate.cc.

Referenced by main().

8.74.2.4 clMPIProc MPIProc

Definition at line 41 of file Locate.cc.

Referenced by main().

8.74.2.5 clLocate* Locate

Definition at line 42 of file Locate.cc.

Referenced by main(), MasterSigHandler(), WrapProcessThread(), and WrapServeClientThread().

8.74.2.6 clSubLocate* SubLocate

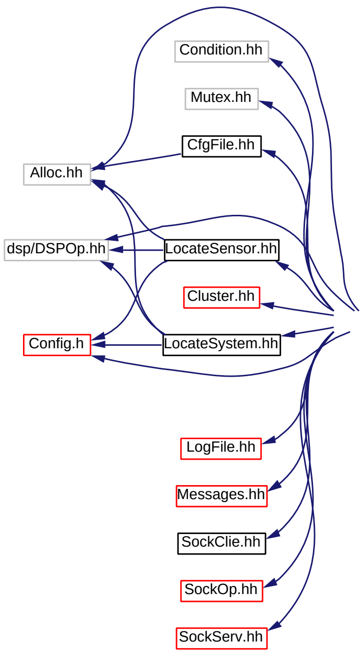
Definition at line 43 of file Locate.cc.

Referenced by main().

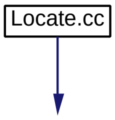
8.75 Locate.hh File Reference

```
#include <Alloc.hh>
#include <Condition.hh>
#include <Mutex.hh>
#include <dsp/DSPOp.hh>
#include "Config.h"
#include "CfgFile.hh"
#include "Cluster.hh"
#include "LocateSensor.hh"
#include "LocateSystem.hh"
#include "LogFile.hh"
#include "Messages.hh"
#include "SockClie.hh"
#include "SockOp.hh"
#include "SockServ.hh"
```

Include dependency graph for Locate.hh:



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



Compounds

- struct **_stDirCfg**

  Configuration information message for subnodes.

- class **clLocate**

  Master node for locate server.

- class **clSubLocate**

  Slave nodes for locate server.

Defines

- #define **LOCATE_HOSTNAME_MAXLEN** 256
- #define **LOCATE_CONTROL_TAG** 1
- #define **LOCATE_PARAM_TAG** 2
- #define **LOCATE_NORMAL_TAG** 3
- #define **LOCATE_CTRL_ACK** 0
- #define **LOCATE_CTRL_STOP** 1

Typedefs

- typedef **_stDirCfg stDirCfg**

  Configuration information message for subnodes.

- typedef **_stDirCfg** ∗ **stpDirCfg**

  Configuration information message for subnodes.

8.75.1 Define Documentation

8.75.1.1 #define LOCATE_HOSTNAME_MAXLEN 256

Definition at line 43 of file Locate.hh.

Referenced by clLocate::Initialize(), clSubLocate::RecvParams(), and clLocate::SendParams().

8.75.1.2 #define LOCATE_CONTROL_TAG 1

Definition at line 44 of file Locate.hh.

Referenced by clLocate::clLocate(), and clSubLocate::clSubLocate().

8.75.1.3 #define LOCATE_PARAM_TAG 2

Definition at line 45 of file Locate.hh.

Referenced by clLocate::clLocate(), clSubLocate::clSubLocate(), and clSubLocate::RecvParams().

8.75.1.4 #define LOCATE_NORMAL_TAG 3

Definition at line 46 of file Locate.hh.

Referenced by clLocate::clLocate(), and clSubLocate::clSubLocate().

8.75.1.5 #define LOCATE_CTRL_ACK 0

Definition at line 48 of file Locate.hh.

8.75.1.6 #define LOCATE_CTRL_STOP 1

Definition at line 49 of file Locate.hh.

Referenced by clLocate::Stop().

8.75.2 Typedef Documentation

8.75.2.1 typedef struct _stDirCfg stDirCfg

Configuration information message for subnodes.

8.75.2.2 typedef struct _stDirCfg * stpDirCfg

Configuration information message for subnodes.

8.76 LocateSensor.cc File Reference

`#include <stdio.h>`

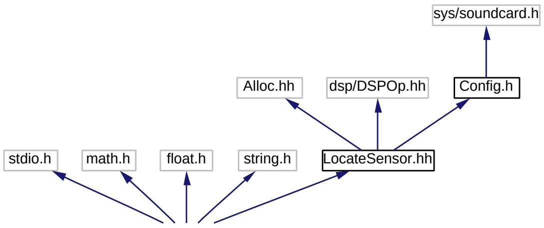
`#include <math.h>`

`#include <float.h>`

`#include <string.h>`

`#include "LocateSensor.hh"`

Include dependency graph for LocateSensor.cc:



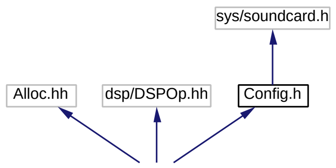
8.77 LocateSensor.hh File Reference

`#include <Alloc.hh>`

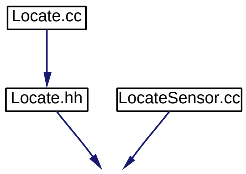
`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

Include dependency graph for LocateSensor.hh:



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



Compounds

- class **clLocateSensor**

  Sensor matrix processing for locating sound sources.

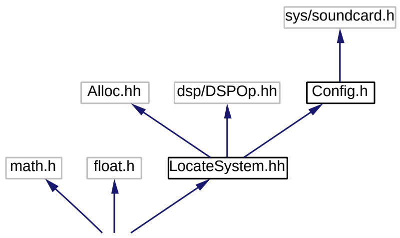
8.78 LocateSystem.cc File Reference

`#include <math.h>`

`#include <float.h>`

`#include "LocateSystem.hh"`

Include dependency graph for LocateSystem.cc:



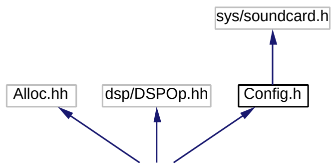
8.79 LocateSystem.hh File Reference

`#include <Alloc.hh>`

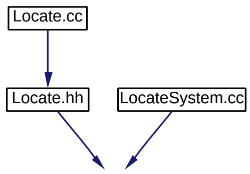
`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

Include dependency graph for LocateSystem.hh:



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



Compounds

- class **clLocateSystem**

  Result combining of locate matrixes.

Defines

- #define **LOCATESYSTEM_HH**

8.79.1 Define Documentation

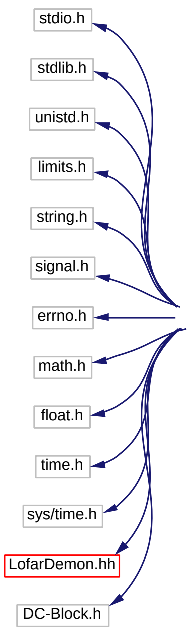
8.79.1.1 #define LOCATESYSTEM_HH

Definition at line 30 of file LocateSystem.hh.

8.80 LofarDemon.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <limits.h>
#include <string.h>
#include <signal.h>
#include <errno.h>
#include <math.h>
#include <float.h>
#include <time.h>
#include <sys/time.h>
#include "LofarDemon.hh"
#include "DC-Block.h"
```

Include dependency graph for LofarDemon.cc:



Defines

- #define **LOFAR_NWINS** 12

Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char *argv[])

Variables

- bool **bDebug**
- bool **bDaemonProc**
- **clLofarDemon** * **LofarDemon**
- const char * **cpaWindowFuncs** []

8.80.1 Define Documentation

8.80.1.1 #define LOFAR_NWINS 12

Definition at line 43 of file LofarDemon.cc.

Referenced by clLofarDemon::InitFFT().

8.80.2 Function Documentation

8.80.2.1 void SigHandler (int *iSigNum*)

Definition at line 59 of file LofarDemon.cc.

References bDebug.

8.80.2.2 int main (int *argc*, char * *argv*[])

Definition at line 75 of file LofarDemon.cc.

References bDaemonProc, bDebug, clLofarDemon::Exec(), LofarDemon, and SigHandler().

8.80.3 Variable Documentation

8.80.3.1 bool bDebug `[static]`

Definition at line 39 of file LofarDemon.cc.

Referenced by clLofarDemon::Exec(), clLofarDemon::InitFFT(), clLofarDemon::InitFilter(), main(), clLofarDemon::MainLoop(), clLofarDemon::SendResults(), and SigHandler().

8.80.3.2 bool bDaemonProc [static]

Definition at line 40 of file LofarDemon.cc.

Referenced by main().

8.80.3.3 clLofarDemon* LofarDemon

Definition at line 41 of file LofarDemon.cc.

Referenced by main().

8.80.3.4 const char* cpaWindowFuncs[] [static]

Initial value:

```
{
   "Rectangle",
   "Bartlett",
   "Blackman",
   "Blackman-Harris",
   "Cosine tapered",
   "Exponential",
   "Flat-top",
   "Hamming",
   "Hanning",
   "Kaiser",
   "Kaiser-Bessel",
   "Tukey" }
```

Definition at line 44 of file LofarDemon.cc.

Referenced by clLofarDemon::InitFFT().

8.81 LofarDemon.hh File Reference

#include <Alloc.hh>

#include <dsp/DSPOp.hh>

#include <dsp/Filter.hh>

#include <dsp/ReBuffer.hh>

#include <dsp/RecDecimator.hh>

#include <dsp/IIRCascade.hh>

#include <dsp/FlipBand.hh>

#include "Config.h"

#include "LocalMsg.h"

#include "CfgFile.hh"

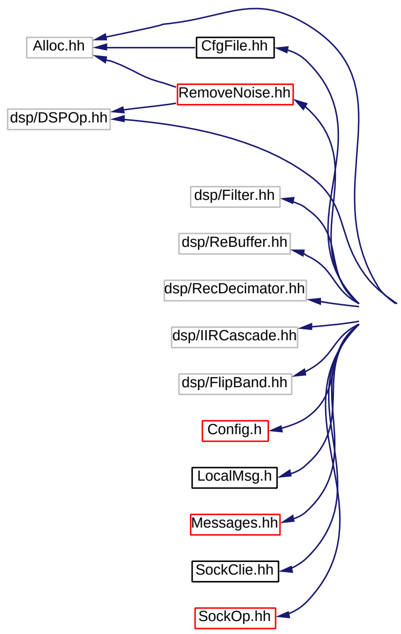
#include "Messages.hh"

#include "RemoveNoise.hh"

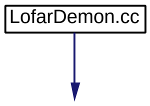
#include "SockClie.hh"

#include "SockOp.hh"

Include dependency graph for LofarDemon.hh:



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



Compounds

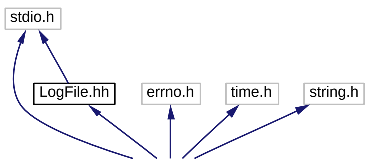
- class **clLofarDemon**

  LOFAR/DEMON calculation server.

8.82 LogFile.cc File Reference

```
#include <stdio.h>
#include <errno.h>
#include <time.h>
#include <string.h>
#include "LogFile.hh"
```

Include dependency graph for LogFile.cc:

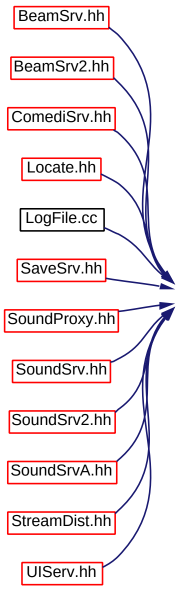


8.83 LogFile.hh File Reference

`#include <stdio.h>`

Include dependency graph for LogFile.hh:

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



Compounds

- class **clLogFile**

  Log file class.

Defines

- #define **LOGFILE_NOERROR** 0

8.83.1 Define Documentation

8.83.1.1 #define LOGFILE_NOERROR 0

Definition at line 29 of file LogFile.hh.

Referenced by clLogFile::clLogFile(), and clUIServer::clUIServer().

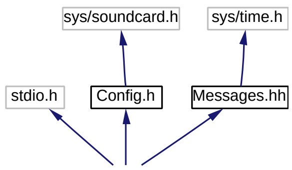
8.84 Messages.cc File Reference

`#include <stdio.h>`

`#include "Config.h"`

`#include "Messages.hh"`

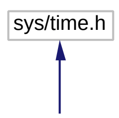
Include dependency graph for Messages.cc:



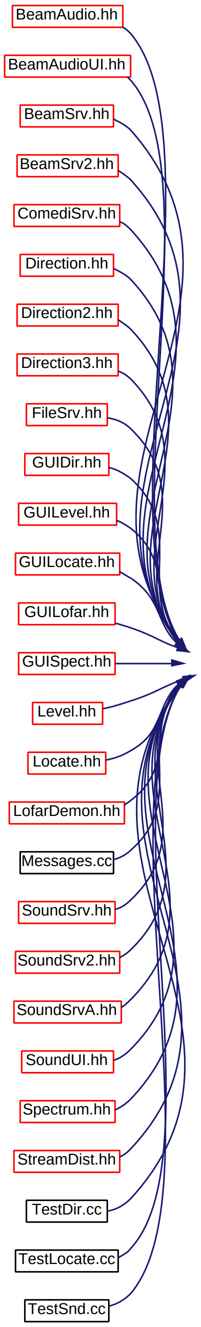
8.85 Messages.hh File Reference

`#include <sys/time.h>`

Include dependency graph for Messages.hh:



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



Compounds

- struct **_stSoundStart**

    Header message from soundsrv.

- struct **_stSpectReq**

  Spectrum: request.

- struct **_stSpectRes**

  Spectrum: result.

- struct **_stDirReq**

  Direction: request.

- struct **_stDirRes**

  Direction: result.

- struct **_stDirReq2**

  Direction3: request.

- struct **_stDirRes2**

  Direction3: result.

- struct **_stLofarReq**

  LOFAR: request.

- struct **_stLofarRes**

  LOFAR: result.

- struct **_stBeamAudioReq**

  BeamAudio: request.

- struct **_stBeamAudioFirst**

  BeamAudio: result header.

- struct **_stBeamAudioRes**

  BeamAudio: result.

- struct **_stLevelReq**

  Level: request.

- struct **_stLevelRes**

  Level: result.

- struct **_stLocateHdr**

  Locate: results header.

- struct **_stLocateRes**

  Locate: result.

- class **clBaseMsg**

Base class for all message handling.

- class **clSoundMsg**

  Sound server communication.

- class **clSpectMsg**

  Spectrum server communication.

- class **clDirMsg**

  Direction server communication.

- class **clDirMsg2**

  Direction server communication 2 (for locate/direction3).

- class **clLofarMsg**

  LOFAR/DEMON server communication.

- class **clBeamAudioMsg**

  BeamAudio server communication.

- class **clLevelMsg**

  Level server communication.

- class **clLocateMsg**

  Locate server communication.

Defines

- #define **MSG_FALSE** 0
- #define **MSG_TRUE** 1

Typedefs

- typedef **_stSoundStart stSoundStart**

  Header message from soundsrv.

- typedef **_stSoundStart** * **stpSoundStart**

  Header message from soundsrv.

- typedef **_stSpectReq stSpectReq**

  Spectrum: request.

- typedef **_stSpectReq** * **stpSpectReq**

  Spectrum: request.

- typedef **_stSpectRes stSpectRes**
  Spectrum: result.
- typedef **_stSpectRes** ∗ **stpSpectRes**
  Spectrum: result.
- typedef **_stDirReq stDirReq**
  Direction: request.
- typedef **_stDirReq** ∗ **stpDirReq**
  Direction: request.
- typedef **_stDirRes stDirRes**
  Direction: result.
- typedef **_stDirRes** ∗ **stpDirRes**
  Direction: result.
- typedef **_stDirReq2 stDirReq2**
  Direction3: request.
- typedef **_stDirReq2** ∗ **stpDirReq2**
  Direction3: request.
- typedef **_stDirRes2 stDirRes2**
  Direction3: result.
- typedef **_stDirRes2** ∗ **stpDirRes2**
  Direction3: result.
- typedef **_stLofarReq stLofarReq**
  LOFAR: request.
- typedef **_stLofarReq** ∗ **stpLofarReq**
  LOFAR: request.
- typedef **_stLofarRes stLofarRes**
  LOFAR: result.
- typedef **_stLofarRes** ∗ **stpLofarRes**
  LOFAR: result.
- typedef **_stBeamAudioReq stBeamAudioReq**
  BeamAudio: request.
- typedef **_stBeamAudioReq** ∗ **stpBeamAudioReq**

BeamAudio: request.

- typedef **_stBeamAudioFirst stBeamAudioFirst**
  BeamAudio: result header.
- typedef **_stBeamAudioFirst** ∗ **stpBeamAudioFirst**
  BeamAudio: result header.
- typedef **_stBeamAudioRes stBeamAudioRes**
  BeamAudio: result.
- typedef **_stBeamAudioRes** ∗ **stpBeamAudioRes**
  BeamAudio: result.
- typedef **_stLevelReq stLevelReq**
  Level: request.
- typedef **_stLevelReq** ∗ **stpLevelReq**
  Level: request.
- typedef **_stLevelRes stLevelRes**
  Level: result.
- typedef **_stLevelRes** ∗ **stpLevelRes**
  Level: result.
- typedef **_stLocateHdr stLocateHdr**
  Locate: results header.
- typedef **_stLocateHdr** ∗ **stpLocateHdr**
  Locate: results header.
- typedef **_stLocateRes stLocateRes**
  Locate: result.
- typedef **_stLocateRes** ∗ **stpLocateRes**
  Locate: result.

Enumerations

- enum { **MSG_SOUND_COMPRESS_NONE** = 0, **MSG_SOUND_-COMPRESS_FLAC** = 1 }
  Sound: compression.

- enum {

  MSG_SPECT_TYPE_FFT = 0, **MSG_SPECT_TYPE_MRFFT** = 1, **MSG_SPECT_TYPE_GABOR** = 2, **MSG_SPECT_TYPE_-WVD** = 3,

  MSG_SPECT_TYPE_HANKEL = 4, **MSG_SPECT_TYPE_-AUTOCORR** = 5, **MSG_SPECT_TYPE_CEPSTRUM** = 6 }

  Spectrum: type.

- enum {

  MSG_SPECT_WND_RECT = 0, **MSG_SPECT_WND_-BARTLETT** = 1, **MSG_SPECT_WND_BLACKMAN** = 2, **MSG_SPECT_WND_BM_HAR** = 3,

  MSG_SPECT_WND_COS_TAPER = 4, **MSG_SPECT_WND_-EXP** = 5, **MSG_SPECT_WND_FLATTOP** = 6, **MSG_SPECT_-WND_GEN_COS** = 7,

  MSG_SPECT_WND_HAMMING = 8, **MSG_SPECT_WND_-HANNING** = 9, **MSG_SPECT_WND_KAISER** = 10, **MSG_-SPECT_WND_KAI_BES** = 11,

  MSG_SPECT_WND_TUKEY = 12 }

  Spectrum: window.

- enum {

  MSG_SPECT_BNER_NONE = 0, **MSG_SPECT_BNER_TPSW** = 1, **MSG_SPECT_BNER_OTA** = 2, **MSG_SPECT_BNER_DIFF** = 3,

  MSG_SPECT_BNER_IDIFF = 4 }

  Spectrum: background noise estimation and removal.

- enum { **MSG_DIR_ALG_BEAM** = 1, **MSG_DIR_ALG_CORR** = 2, **MSG_DIR_ALG_SPECT** = 4 }

  Direction: algorithm.

- enum { **MSG_DIR_SCAL_LIN** = 0, **MSG_DIR_SCAL_LOG** = 1, **MSG_DIR_SCAL_EXP** = 2, **MSG_DIR_SCAL_SIN** = 3 }

  Direction: scaling.

- enum {

  MSG_DIR_BNER_NONE = 0, **MSG_DIR_BNER_TPSW** = 1, **MSG_DIR_BNER_OTA** = 2, **MSG_DIR_BNER_DIFF** = 3,

  MSG_DIR_BNER_IDIFF = 4 }

  Direction: background noise estimation and removal.

- enum { **MSG_LOFAR_TYPE_FFT** = 0, **MSG_LOFAR_TYPE_-CEPSTRUM** = 1, **MSG_LOFAR_TYPE_AUTOCORR** = 2 }

  LOFAR: type.

- enum {

  MSG_LOFAR_WIN_RECT = 0, **MSG_LOFAR_WIN_BARTLETT** = 1, **MSG_LOFAR_WIN_BLACKMAN** = 2, **MSG_LOFAR_WIN_BM_HARRIS** = 3,

  MSG_LOFAR_WIN_COS_TAPER = 4, **MSG_LOFAR_WIN_EXP** = 5, **MSG_LOFAR_WIN_FLATTOP** = 6, **MSG_LOFAR_WIN_HAMMING** = 7,

  MSG_LOFAR_WIN_HANNING = 8, **MSG_LOFAR_WIN_KAISER** = 9, **MSG_LOFAR_WIN_KAI_BES** = 10, **MSG_LOFAR_WIN_TUKEY** = 11 }

  LOFAR: window.

- enum {

  MSG_LOFAR_BNER_NONE = 0, **MSG_LOFAR_BNER_TPSW** = 1, **MSG_LOFAR_BNER_OTA** = 2, **MSG_LOFAR_BNER_DIFF** = 3,

  MSG_LOFAR_BNER_IDIFF = 4, **MSG_LOFAR_BNER_STDDEV** = 5 }

  LOFAR: background noise estimation and removal.

- enum {

  MSG_LOFAR_CLIP_NONE = 0, **MSG_LOFAR_CLIP_LOW** = 1, **MSG_LOFAR_CLIP_BOTH** = 2, **MSG_LOFAR_CLIP_MEAN** = 3,

  MSG_LOFAR_CLIP_MEDIAN = 4, **MSG_LOFAR_CLIP_10DB** = 5, **MSG_LOFAR_CLIP_20DB** = 6, **MSG_LOFAR_CLIP_50P** = 7,

  MSG_LOFAR_CLIP_75P = 8, **MSG_LOFAR_CLIP_OFFSET** = 9, **MSG_LOFAR_CLIP_OFFSET2** = 10, **MSG_LOFAR_CLIP_OFFSET3** = 11,

  MSG_LOFAR_CLIP_SLIDING = 12 }

  LOFAR: predefined clipping functions.

- enum {

  MSG_LEVEL_ALG_PEAK = 0, **MSG_LEVEL_ALG_RMS** = 1, **MSG_LEVEL_ALG_MEAN** = 2, **MSG_LEVEL_ALG_MEDIAN** = 3,

  MSG_LEVEL_ALG_STDDEV = 4 }

  Level: algorithm.

8.85.1 Define Documentation

8.85.1.1 #define MSG_FALSE 0

Definition at line 29 of file Messages.hh.

Referenced by clLofarMsg::SetRequest(), clSpectMsg::SetRequest(), clDirMsg::SetResult(), and clSpectMsg::SetResult().

8.85.1.2 #define MSG_TRUE 1

Definition at line 30 of file Messages.hh.

Referenced by clBeamAudioMsg::GetRequest(), clLofarMsg::GetRequest(), clDirMsg::GetRequest(), clSpectMsg::GetRequest(), clDirMsg::GetResult(), clSpectMsg::GetResult(), clLofarMsg::SetRequest(), clSpectMsg::SetRequest(), clDirMsg::SetResult(), and clSpectMsg::SetResult().

8.85.2 Typedef Documentation

8.85.2.1 typedef struct _stSoundStart stSoundStart

Header message from soundsrv.

8.85.2.2 typedef struct _stSoundStart * stpSoundStart

Header message from soundsrv.

8.85.2.3 typedef struct _stSpectReq stSpectReq

Spectrum: request.

8.85.2.4 typedef struct _stSpectReq * stpSpectReq

Spectrum: request.

8.85.2.5 typedef struct _stSpectRes stSpectRes

Spectrum: result.

8.85.2.6 typedef struct _stSpectRes * stpSpectRes

Spectrum: result.

8.85.2.7 typedef struct _stDirReq stDirReq

Direction: request.

8.85.2.8 typedef struct _stDirReq * stpDirReq

Direction: request.

8.85.2.9 typedef struct _stDirRes stDirRes

Direction: result.

8.85.2.10 typedef struct _stDirRes * stpDirRes

Direction: result.

8.85.2.11 typedef struct _stDirReq2 stDirReq2

Direction3: request.

This is used by locate server.

8.85.2.12 typedef struct _stDirReq2 * stpDirReq2

Direction3: request.

This is used by locate server.

8.85.2.13 typedef struct _stDirRes2 stDirRes2

Direction3: result.

This is used by locate server.

8.85.2.14 typedef struct _stDirRes2 * stpDirRes2

Direction3: result.

This is used by locate server.

8.85.2.15 typedef struct _stLofarReq stLofarReq

LOFAR: request.

8.85.2.16 typedef struct _stLofarReq * stpLofarReq

LOFAR: request.

8.85.2.17 typedef struct _stLofarRes stLofarRes

LOFAR: result.

8.85.2.18 typedef struct _stLofarRes * stpLofarRes

LOFAR: result.

8.85.2.19 typedef struct _stBeamAudioReq stBeamAudioReq

BeamAudio: request.

8.85.2.20 typedef struct _stBeamAudioReq * stpBeamAudioReq

BeamAudio: request.

8.85.2.21 typedef struct _stBeamAudioFirst stBeamAudioFirst

BeamAudio: result header.

8.85.2.22 typedef struct _stBeamAudioFirst * stpBeamAudioFirst

BeamAudio: result header.

8.85.2.23 typedef struct _stBeamAudioRes stBeamAudioRes

BeamAudio: result.

8.85.2.24 typedef struct _stBeamAudioRes * stpBeamAudioRes

BeamAudio: result.

8.85.2.25 typedef struct _stLevelReq stLevelReq

Level: request.

8.85.2.26 typedef struct _stLevelReq * stpLevelReq

Level: request.

8.85.2.27 typedef struct _stLevelRes stLevelRes

Level: result.

8.85.2.28 typedef struct _stLevelRes * stpLevelRes

Level: result.

8.85.2.29 typedef struct _stLocateHdr stLocateHdr

Locate: results header.

8.85.2.30 typedef struct _stLocateHdr * stpLocateHdr

Locate: results header.

8.85.2.31 typedef struct _stLocateRes stLocateRes

Locate: result.

8.85.2.32 typedef struct _stLocateRes * stpLocateRes

Locate: result.

8.85.3 Enumeration Type Documentation

8.85.3.1 anonymous enum

Sound: compression.

Enumeration values:
- **MSG_SOUND_COMPRESS_NONE**
- **MSG_SOUND_COMPRESS_FLAC**

Definition at line 36 of file Messages.hh.

8.85.3.2 anonymous enum

Spectrum: type.

Enumeration values:
- **MSG_SPECT_TYPE_FFT**
- **MSG_SPECT_TYPE_MRFFT**
- **MSG_SPECT_TYPE_GABOR**

MSG_SPECT_TYPE_WVD

MSG_SPECT_TYPE_HANKEL

MSG_SPECT_TYPE_AUTOCORR

MSG_SPECT_TYPE_CEPSTRUM

Definition at line 45 of file Messages.hh.

8.85.3.3 anonymous enum

Spectrum: window.

Enumeration values:

MSG_SPECT_WND_RECT

MSG_SPECT_WND_BARTLETT

MSG_SPECT_WND_BLACKMAN

MSG_SPECT_WND_BM_HAR

MSG_SPECT_WND_COS_TAPER

MSG_SPECT_WND_EXP

MSG_SPECT_WND_FLATTOP

MSG_SPECT_WND_GEN_COS

MSG_SPECT_WND_HAMMING

MSG_SPECT_WND_HANNING

MSG_SPECT_WND_KAISER

MSG_SPECT_WND_KAI_BES

MSG_SPECT_WND_TUKEY

Definition at line 59 of file Messages.hh.

8.85.3.4 anonymous enum

Spectrum: background noise estimation and removal.

Enumeration values:

MSG_SPECT_BNER_NONE

MSG_SPECT_BNER_TPSW

MSG_SPECT_BNER_OTA

MSG_SPECT_BNER_DIFF

MSG_SPECT_BNER_IDIFF

Definition at line 79 of file Messages.hh.

8.85.3.5 anonymous enum

Direction: algorithm.

Enumeration values:
- **MSG_DIR_ALG_BEAM**
- **MSG_DIR_ALG_CORR**
- **MSG_DIR_ALG_SPECT**

Definition at line 91 of file Messages.hh.

8.85.3.6 anonymous enum

Direction: scaling.

Enumeration values:
- **MSG_DIR_SCAL_LIN**
- **MSG_DIR_SCAL_LOG**
- **MSG_DIR_SCAL_EXP**
- **MSG_DIR_SCAL_SIN**

Definition at line 101 of file Messages.hh.

8.85.3.7 anonymous enum

Direction: background noise estimation and removal.

Enumeration values:
- **MSG_DIR_BNER_NONE**
- **MSG_DIR_BNER_TPSW**
- **MSG_DIR_BNER_OTA**
- **MSG_DIR_BNER_DIFF**
- **MSG_DIR_BNER_IDIFF**

Definition at line 112 of file Messages.hh.

8.85.3.8 anonymous enum

LOFAR: type.

Enumeration values:
- **MSG_LOFAR_TYPE_FFT**
- **MSG_LOFAR_TYPE_CEPSTRUM**
- **MSG_LOFAR_TYPE_AUTOCORR**

Definition at line 124 of file Messages.hh.

8.85.3.9 anonymous enum

LOFAR: window.

Enumeration values:

MSG_LOFAR_WIN_RECT

MSG_LOFAR_WIN_BARTLETT

MSG_LOFAR_WIN_BLACKMAN

MSG_LOFAR_WIN_BM_HARRIS

MSG_LOFAR_WIN_COS_TAPER

MSG_LOFAR_WIN_EXP

MSG_LOFAR_WIN_FLATTOP

MSG_LOFAR_WIN_HAMMING

MSG_LOFAR_WIN_HANNING

MSG_LOFAR_WIN_KAISER

MSG_LOFAR_WIN_KAI_BES

MSG_LOFAR_WIN_TUKEY

Definition at line 134 of file Messages.hh.

8.85.3.10 anonymous enum

LOFAR: background noise estimation and removal.

Enumeration values:

MSG_LOFAR_BNER_NONE

MSG_LOFAR_BNER_TPSW

MSG_LOFAR_BNER_OTA

MSG_LOFAR_BNER_DIFF

MSG_LOFAR_BNER_IDIFF

MSG_LOFAR_BNER_STDDEV

Definition at line 153 of file Messages.hh.

8.85.3.11 anonymous enum

LOFAR: predefined clipping functions.

Enumeration values:

MSG_LOFAR_CLIP_NONE

MSG_LOFAR_CLIP_LOW

MSG_LOFAR_CLIP_BOTH

MSG_LOFAR_CLIP_MEAN
MSG_LOFAR_CLIP_MEDIAN
MSG_LOFAR_CLIP_10DB
MSG_LOFAR_CLIP_20DB
MSG_LOFAR_CLIP_50P
MSG_LOFAR_CLIP_75P
MSG_LOFAR_CLIP_OFFSET
MSG_LOFAR_CLIP_OFFSET2
MSG_LOFAR_CLIP_OFFSET3
MSG_LOFAR_CLIP_SLIDING

Definition at line 166 of file Messages.hh.

8.85.3.12 anonymous enum

Level: algorithm.

Enumeration values:

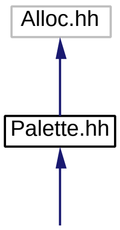
MSG_LEVEL_ALG_PEAK
MSG_LEVEL_ALG_RMS
MSG_LEVEL_ALG_MEAN
MSG_LEVEL_ALG_MEDIAN
MSG_LEVEL_ALG_STDDEV

Definition at line 186 of file Messages.hh.

8.86 Palette.cc File Reference

`#include "Palette.hh"`

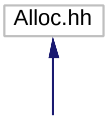
Include dependency graph for Palette.cc:



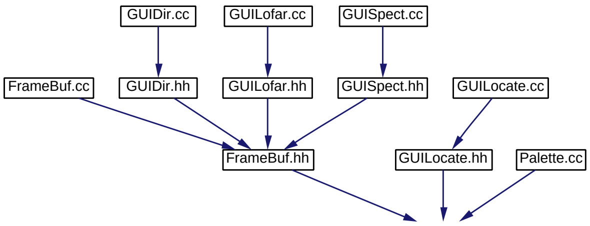
8.87 Palette.hh File Reference

`#include <Alloc.hh>`

Include dependency graph for Palette.hh:



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



Compounds

- class **clPalette**

  RGB palette class.

8.88 RemoveNoise.cc File Reference

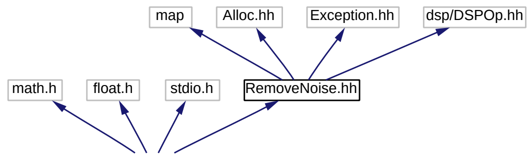
`#include <math.h>`

`#include <float.h>`

`#include <stdio.h>`

`#include "RemoveNoise.hh"`

Include dependency graph for RemoveNoise.cc:



8.89 RemoveNoise.hh File Reference

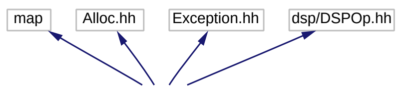
`#include <map>`

`#include <Alloc.hh>`

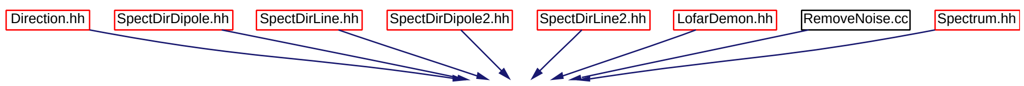
`#include <Exception.hh>`

`#include <dsp/DSPOp.hh>`

Include dependency graph for RemoveNoise.hh:



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



Compounds

- class **clSortedArray**

  Class for array of sorted values with insert/remove functionality.

- class **clSortedArray.clXSortedArray**
- class **clRemoveNoise**

  Background noise estimation and removal.

Typedefs

- typedef std::map< double, long > **ValueMap_t**
- typedef std::map< float, long > **ValueMapf_t**

8.89.1 Typedef Documentation

8.89.1.1 typedef std::map<double, long> ValueMap_t

Definition at line 33 of file RemoveNoise.hh.

8.89.1.2 typedef std::map<float, long> ValueMapf_t

Definition at line 34 of file RemoveNoise.hh.

8.90 SaveSrv.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <limits.h>
#include <errno.h>
#include <time.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sched.h>
#include <sys/types.h>
#include <sys/stat.h>
#include "SockClie.hh"
#include "SaveSrv.hh"
```

Include dependency graph for SaveSrv.cc:

Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char *argv[])

Variables

- bool **bDaemon** = false
- const char * **cpFileExt** [] = { "wav", "aiff", "flac" }
- const char **cpFLACMarker** [] = "fLaC"
- **clSaveSrv SaveSrv**

8.90.1 Function Documentation

8.90.1.1 void SigHandler (int *iSigNum*)

Definition at line 46 of file SaveSrv.cc.

References clLogFile::Add(), clSaveSrv::Log, SaveSrv, and clSaveSrv::Stop().

8.90.1.2 int main (int *argc*, char * *argv*[])

Definition at line 66 of file SaveSrv.cc.

References clSaveSrv::Main(), SaveSrv, and SigHandler().

8.90.2 Variable Documentation

8.90.2.1 bool bDaemon = false [static]

Definition at line 41 of file SaveSrv.cc.

Referenced by clSaveSrv::Main().

8.90.2.2 const char* cpFileExt[ ] = { "wav", "aiff", "flac" } [static]

Definition at line 42 of file SaveSrv.cc.

Referenced by clSaveSrv::CreateFileName().

8.90.2.3 const char cpFLACMarker[ ] = "fLaC" [static]

Definition at line 43 of file SaveSrv.cc.

8.90.2.4 clSaveSrv SaveSrv

Definition at line 44 of file SaveSrv.cc.

Referenced by main(), and SigHandler().

8.91 SaveSrv.hh File Reference

`#include <stdio.h>`

`#include <limits.h>`

`#include <time.h>`

`#include <sndfile.h>`

`#include <Alloc.hh>`

`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

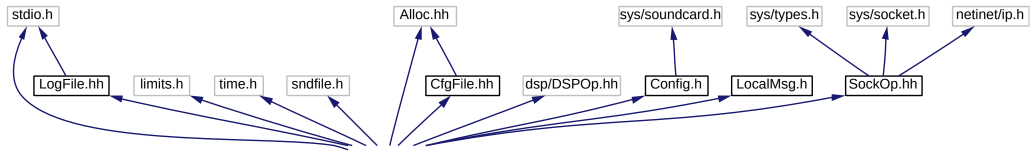
`#include "LocalMsg.h"`

`#include "CfgFile.hh"`

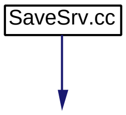
`#include "LogFile.hh"`

`#include "SockOp.hh"`

Include dependency graph for SaveSrv.hh:



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



Compounds

- class **clSaveSrv**

  Save server.

Defines

- #define **SAVS_SNDFILE_MSGLEN** 255
- #define **SAVS_LOGENTRY_SIZE** 255

- #define **SAVS_TIMELEN** 12
- #define **SAVS_DEF_FRAMELEN** 8192
- #define **SAVS_DEF_FILETIME** 60

Enumerations

- enum { **SAVS_FORMAT_WAV** = 0, **SAVS_FORMAT_AIFF** = 1, **SAVS_FORMAT_FLAC** = 2 }

  File format.

- enum { **SAVS_TYPE_PCM** = 0, **SAVS_TYPE_FLOAT** = 1, **SAVS_-TYPE_ADPCM** = 2, **SAVS_TYPE_MSADPCM** = 3 }

  File datatype.

8.91.1 Define Documentation

8.91.1.1 #define SAVS_SNDFILE_MSGLEN 255

Definition at line 47 of file SaveSrv.hh.

Referenced by clSaveSrv::Initialize(), and clSaveSrv::WriteData().

8.91.1.2 #define SAVS_LOGENTRY_SIZE 255

Definition at line 48 of file SaveSrv.hh.

Referenced by clSaveSrv::ConnectStream().

8.91.1.3 #define SAVS_TIMELEN 12

Definition at line 49 of file SaveSrv.hh.

Referenced by clSaveSrv::CreateFileName().

8.91.1.4 #define SAVS_DEF_FRAMELEN 8192

Definition at line 50 of file SaveSrv.hh.

Referenced by clSaveSrv::Initialize().

8.91.1.5 #define SAVS_DEF_FILETIME 60

Definition at line 51 of file SaveSrv.hh.

Referenced by clSaveSrv::Initialize().

8.91.2 Enumeration Type Documentation

8.91.2.1 anonymous enum

File format.

Enumeration values:
- **SAVS_FORMAT_WAV**
- **SAVS_FORMAT_AIFF**
- **SAVS_FORMAT_FLAC**

Definition at line 57 of file SaveSrv.hh.

8.91.2.2 anonymous enum

File datatype.

Enumeration values:
- **SAVS_TYPE_PCM**
- **SAVS_TYPE_FLOAT**
- **SAVS_TYPE_ADPCM**
- **SAVS_TYPE_MSADPCM**

Definition at line 67 of file SaveSrv.hh.

8.92 SockClie.cc File Reference

`#include <stdio.h>`

`#include <stdlib.h>`

`#include <string.h>`

`#include <errno.h>`

`#include <netdb.h>`

`#include <sys/types.h>`

`#include <sys/socket.h>`

`#include <sys/un.h>`

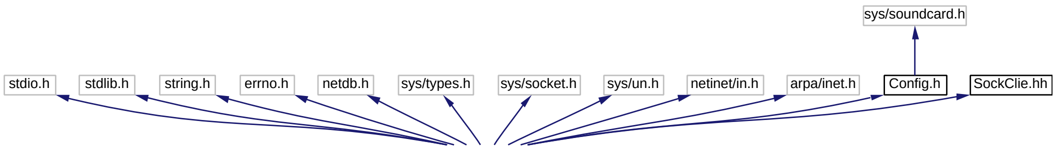
`#include <netinet/in.h>`

`#include <arpa/inet.h>`

`#include "Config.h"`

`#include "SockClie.hh"`

Include dependency graph for SockClie.cc:



Defines

- #define **UNIX_PATH_MAX** 108

8.92.1 Define Documentation

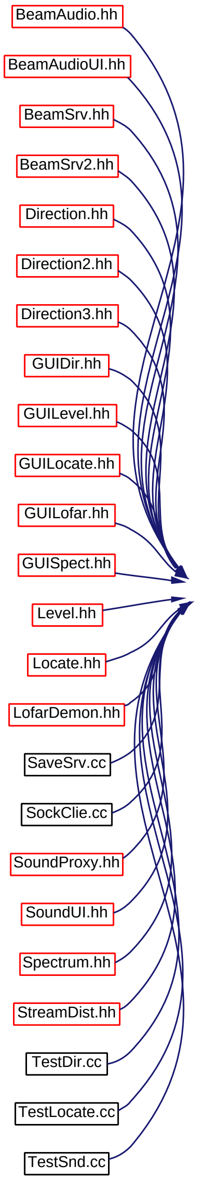
8.92.1.1 #define UNIX_PATH_MAX 108

Definition at line 39 of file SockClie.cc.

Referenced by clSockServ::Bind(), and clSockClie::Connect().

8.93 SockClie.hh File Reference

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



Compounds

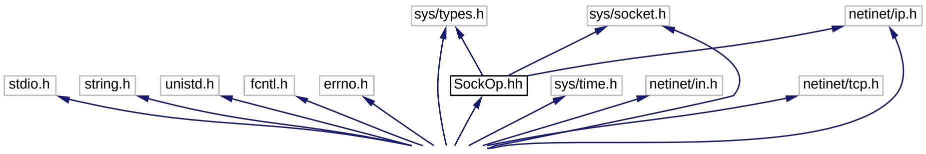
- class **clSockClie**

  Socket client class.

8.94 SockOp.cc File Reference

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/tcp.h>
#include "SockOp.hh"
```

Include dependency graph for SockOp.cc:



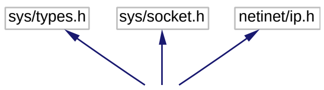
8.95 SockOp.hh File Reference

`#include <sys/types.h>`

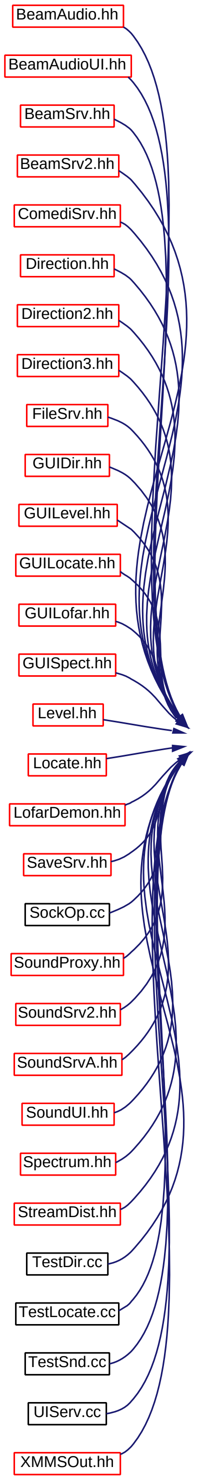
`#include <sys/socket.h>`

`#include <netinet/ip.h>`

Include dependency graph for SockOp.hh:



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


BeamAudio.hh
BeamAudioUI.hh
BeamSrv.hh
BeamSrv2.hh
ComediSrv.hh
Direction.hh
Direction2.hh
Direction3.hh
FileSrv.hh
GUIDir.hh
GUILevel.hh
GUILocate.hh
GUILofar.hh
GUISpect.hh
Level.hh
Locate.hh
LofarDemon.hh
SaveSrv.hh
SockOp.cc
SoundProxy.hh
SoundSrv2.hh
SoundSrvA.hh
SoundUI.hh
Spectrum.hh
StreamDist.hh
TestDir.cc
TestLocate.cc
TestSnd.cc
UIServ.cc
XMMSOut.hh

Compounds

- class **clSockOp**

  Socket I/O operations.

8.96 SockServ.cc File Reference

`#include <stdio.h>`

`#include <stdlib.h>`

`#include <string.h>`

`#include <errno.h>`

`#include <unistd.h>`

`#include <netdb.h>`

`#include <sys/types.h>`

`#include <sys/stat.h>`

`#include <sys/socket.h>`

`#include <sys/un.h>`

`#include <sys/time.h>`

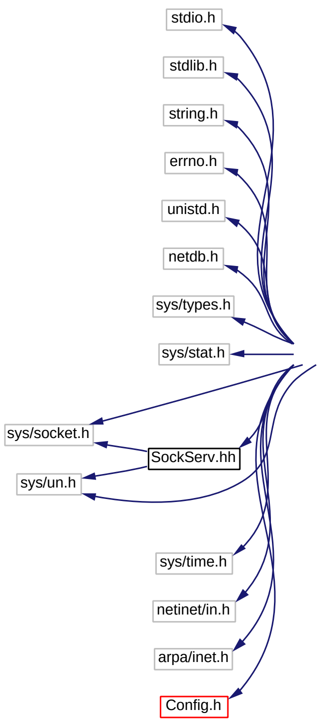
`#include <netinet/in.h>`

`#include <arpa/inet.h>`

`#include "Config.h"`

`#include "SockServ.hh"`

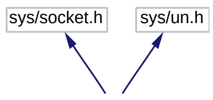
Include dependency graph for SockServ.cc:



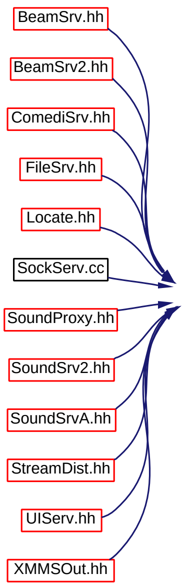
8.97 SockServ.hh File Reference

```
#include <sys/socket.h>
#include <sys/un.h>
```

Include dependency graph for SockServ.hh:



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



Compounds

- class **clSockServ**

  Socket server class.

Defines

- #define **UNIX_PATH_MAX** 108
- #define **SOCKSERV_LISTENQUEUE_LEN** 8

8.97.1 Define Documentation

8.97.1.1 #define UNIX_PATH_MAX 108

Definition at line 31 of file SockServ.hh.

8.97.1.2 #define SOCKSERV_LISTENQUEUE_LEN 8

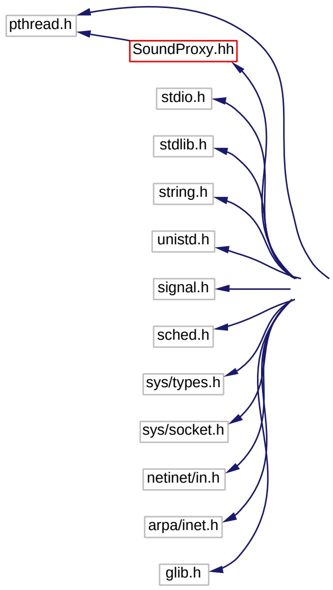
Definition at line 34 of file SockServ.hh.

Referenced by clSockServ::Bind().

8.98 SoundProxy.cc File Reference

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
#include <sched.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <glib.h>
#include "SoundProxy.hh"
```

Include dependency graph for SoundProxy.cc:



Functions

- int **main** (int argc, char *argv[])
- void * **WrapSoundInThread** (void *vpData)
- void * **WrapWaitConnectThread** (void *vpData)

- void * **WrapServeClientThread** (void *vpData)

Variables

- bool **bDaemon**
- bool **bDebug**
- **clSoundProxy** * **SoundProxy**

8.98.1 Function Documentation

8.98.1.1 int main (int *argc*, char * *argv*[ ])

Definition at line 44 of file SoundProxy.cc.

References bDaemon, bDebug, clSoundProxy::Exec(), SoundProxy, SP_VERSMAJ, SP_VERSMIN, and SP_VERSPL.

8.98.1.2 void* WrapSoundInThread (void * *vpData*)

Definition at line 90 of file SoundProxy.cc.

References clSoundProxy::SoundInThread(), and SoundProxy.

Referenced by clSoundProxy::Exec(), and clSoundUI::OnClickedEvent().

8.98.1.3 void* WrapWaitConnectThread (void * *vpData*)

Definition at line 96 of file SoundProxy.cc.

References SoundProxy, and clSoundProxy::WaitConnectThread().

Referenced by clSoundProxy::Exec().

8.98.1.4 void* WrapServeClientThread (void * *vpData*)

Definition at line 102 of file SoundProxy.cc.

References clSoundProxy::ServeClientThread(), and SoundProxy.

8.98.2 Variable Documentation

8.98.2.1 bool bDaemon `[static]`

Definition at line 39 of file SoundProxy.cc.

Referenced by main().

8.98.2.2 bool bDebug [static]

Definition at line 40 of file SoundProxy.cc.

Referenced by main().

8.98.2.3 clSoundProxy* SoundProxy

Definition at line 41 of file SoundProxy.cc.

Referenced by main(), WrapServeClientThread(), WrapSoundInThread(), and WrapWaitConnectThread().

8.99 SoundProxy.hh File Reference

`#include <pthread.h>`

`#include <limits.h>`

`#include <Alloc.hh>`

`#include <Condition.hh>`

`#include <Mutex.hh>`

`#include "Config.h"`

`#include "CfgFile.hh"`

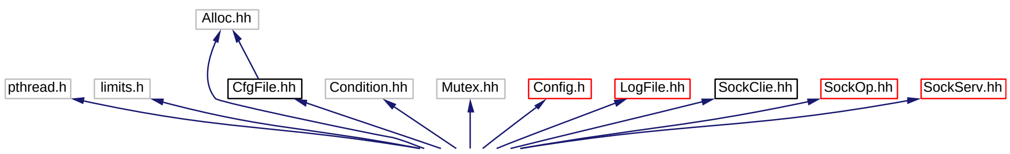
`#include "LogFile.hh"`

`#include "SockClie.hh"`

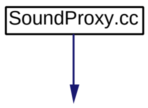
`#include "SockOp.hh"`

`#include "SockServ.hh"`

Include dependency graph for SoundProxy.hh:



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



Compounds

- class **clSoundProxy**

  Sound service proxy.

Defines

- #define **SP_CONV_BUF_LEN** 255

8.99.1 Define Documentation

8.99.1.1 #define SP_CONV_BUF_LEN 255

Definition at line 41 of file SoundProxy.hh.

Referenced by clSoundProxy::ServeClientThread(), and clSoundProxy::SoundInThread().

8.100 SoundSrv.cc File Reference

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <limits.h>

#include <errno.h>

#include <signal.h>

#include <time.h>

#include <sched.h>

#include <netdb.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <sys/mman.h>

#include <netinet/in.h>

#include <netinet/ip.h>

#include <netinet/tcp.h>

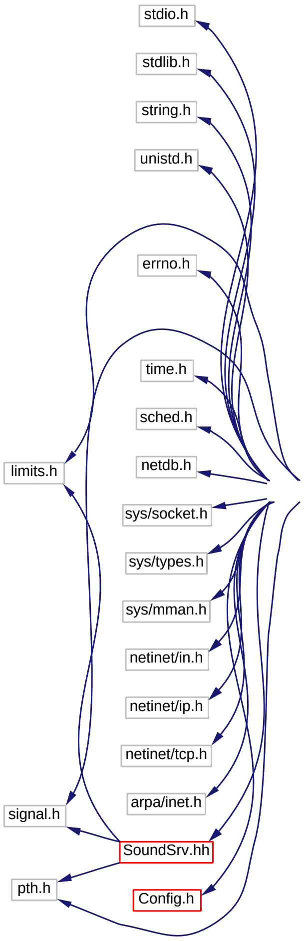
#include <arpa/inet.h>

#include <pth.h>

#include "Config.h"

#include "SoundSrv.hh"

Include dependency graph for SoundSrv.cc:



Functions

- void ∗ **Input** (void ∗vpNone)
- void ∗ **WaitConnect** (void ∗vpNone)
- void ∗ **ServeClient** (void ∗vpHandle)
- int **main** (int argc, char ∗argv[])

Variables

- **clSoundSrv** ∗ **SoundSrv**

8.100.1 Function Documentation

8.100.1.1 void* Input (void * *vpNone*)

Definition at line 50 of file SoundSrv.cc.

References clSoundSrv::InputExec(), and SoundSrv.

Referenced by clSoundSrv::Exec().

8.100.1.2 void* WaitConnect (void * *vpNone*)

Definition at line 57 of file SoundSrv.cc.

References SoundSrv, and clSoundSrv::WaitConnectExec().

Referenced by clSoundSrv::Exec().

8.100.1.3 void* ServeClient (void * *vpHandle*)

Definition at line 64 of file SoundSrv.cc.

References clSoundSrv::ServeClientExec(), and SoundSrv.

Referenced by clSoundSrv::WaitConnectExec().

8.100.1.4 int main (int *argc*, char * *argv*[ ])

Definition at line 71 of file SoundSrv.cc.

References clSoundSrv::Exec(), SoundSrv, SS_SCHED_PRIORITY, SS_SND_DEVICE, SS_VERSMAJ, SS_VERSMIN, and SS_VERSPL.

8.100.2 Variable Documentation

8.100.2.1 clSoundSrv* SoundSrv

Definition at line 47 of file SoundSrv.cc.

Referenced by Input(), clSoundSrv::InputExec(), main(), ServeClient(), and WaitConnect().

8.101 SoundSrv.hh File Reference

`#include <limits.h>`

`#include <signal.h>`

`#include <pth.h>`

`#include <PthMutex.hh>`

`#include <PthCond.hh>`

`#include <dsp/DSPOp.hh>`

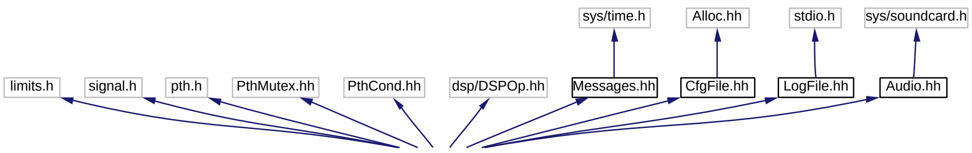
`#include "Messages.hh"`

`#include "CfgFile.hh"`

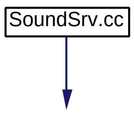
`#include "LogFile.hh"`

`#include "Audio.hh"`

Include dependency graph for SoundSrv.hh:



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



Compounds

- class **clSoundSrv**

  Sound card input server.

Defines

- #define **SS_PTH_MAJ** ((PTH_VERSION&0xf00000)>>20)
- #define **SS_PTH_MIN** ((PTH_VERSION&0x0ff000)>>12)
- #define **SS_PTH_PL** (PTH_VERSION&0x0000ff)

8.101.1 Define Documentation

8.101.1.1 #define SS_PTH_MAJ ((PTH_VERSION&0xf00000)>>20)

Definition at line 40 of file SoundSrv.hh.

Referenced by clSoundSrv::clSoundSrv().

8.101.1.2 #define SS_PTH_MIN ((PTH_VERSION&0x0ff000)>>12)

Definition at line 41 of file SoundSrv.hh.

Referenced by clSoundSrv::clSoundSrv().

8.101.1.3 #define SS_PTH_PL (PTH_VERSION&0x0000ff)

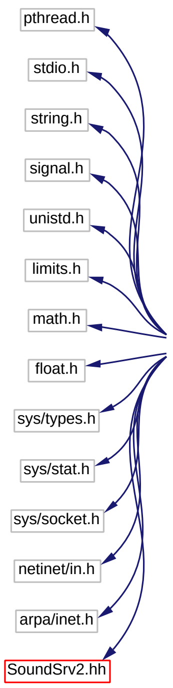
Definition at line 42 of file SoundSrv.hh.

Referenced by clSoundSrv::clSoundSrv().

8.102 SoundSrv2.cc File Reference

```
#include <pthread.h>
#include <stdio.h>
#include <string.h>
#include <signal.h>
#include <unistd.h>
#include <limits.h>
#include <math.h>
#include <float.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include "SoundSrv2.hh"
```

Include dependency graph for SoundSrv2.cc:



Functions

- int **main** (int argc, char *argv[ ])
- void **SigHandler** (int iSigNum)
- void * **WrapAudioInThread** (void *vpParam)
- void * **WrapServeClientThread** (void *vpParam)

Variables

- bool **bDaemon** = false
- **clSoundSrv2 SoundSrv2**

8.102.1 Function Documentation

8.102.1.1 int main (int *argc*, char * *argv*[ ])

Definition at line 44 of file SoundSrv2.cc.

References clSoundSrv2::Main(), and SoundSrv2.

8.102.1.2 void SigHandler (int *iSigNum*)

Definition at line 50 of file SoundSrv2.cc.

References clLogFile::Add(), clSoundSrv2::Log, SoundSrv2, and clSoundSrv2::Stop().

8.102.1.3 void* WrapAudioInThread (void * *vpParam*)

Definition at line 73 of file SoundSrv2.cc.

References clSoundSrv2::AudioInThread(), and SoundSrv2.

8.102.1.4 void* WrapServeClientThread (void * *vpParam*)

Definition at line 79 of file SoundSrv2.cc.

References clSoundSrv2::ServeClientThread(), and SoundSrv2.

8.102.2 Variable Documentation

8.102.2.1 bool bDaemon = false [static]

Definition at line 40 of file SoundSrv2.cc.

Referenced by clSoundSrv2::Main().

8.102.2.2 clSoundSrv2 SoundSrv2

Definition at line 41 of file SoundSrv2.cc.

Referenced by main(), SigHandler(), WrapAudioInThread(), and WrapServeClientThread().

8.103 SoundSrv2.hh File Reference

`#include <Alloc.hh>`

`#include <Condition.hh>`

`#include <Mutex.hh>`

`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

`#include "Audio.hh"`

`#include "CfgFile.hh"`

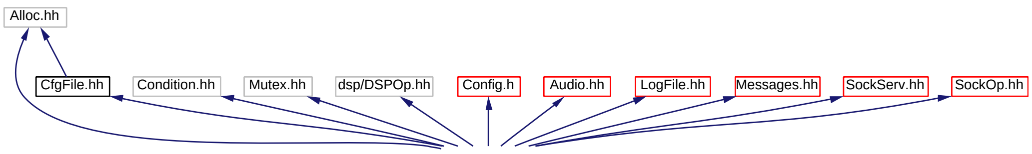
`#include "LogFile.hh"`

`#include "Messages.hh"`

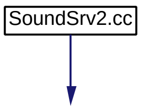
`#include "SockServ.hh"`

`#include "SockOp.hh"`

Include dependency graph for SoundSrv2.hh:



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



Compounds

- class **clSoundSrv2**

  SoundServer2.

Defines

- #define **SS2_LOGENTRY_SIZE** 256
- #define **SS2_OSS_MAJOR**(x) ((x >> 16) & 0xff)

- #define **SS2_OSS_MINOR**(x) ((x >> 8) & 0xff)
- #define **SS2_OSS_PL**(x) (x & 0xff)

8.103.1 Define Documentation

8.103.1.1 #define SS2_LOGENTRY_SIZE 256

Definition at line 46 of file SoundSrv2.hh.

Referenced by clSoundSrv2::AudioInThread(), and clSoundSrv2::ServeClient-Thread().

8.103.1.2 #define SS2_OSS_MAJOR(x) ((x >> 16) & 0xff)

Definition at line 48 of file SoundSrv2.hh.

Referenced by clSoundSrv2::AudioInThread().

8.103.1.3 #define SS2_OSS_MINOR(x) ((x >> 8) & 0xff)

Definition at line 49 of file SoundSrv2.hh.

Referenced by clSoundSrv2::AudioInThread().

8.103.1.4 #define SS2_OSS_PL(x) (x & 0xff)

Definition at line 50 of file SoundSrv2.hh.

Referenced by clSoundSrv2::AudioInThread().

8.104 SoundSrvA.cc File Reference

#include <pthread.h>

#include <stdio.h>

#include <string.h>

#include <signal.h>

#include <unistd.h>

#include <limits.h>

#include <sys/types.h>

#include <sys/stat.h>

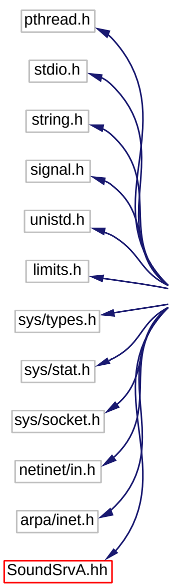
#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include "SoundSrvA.hh"

Include dependency graph for SoundSrvA.cc:



Functions

- int **main** (int argc, char *argv[])
- void **SigHandler** (int iSigNum)
- void * **WrapAudioInThread** (void *vpParam)
- void * **WrapServeClientThread** (void *vpParam)

Variables

- bool **bDaemon** = false
- **clSoundSrvA SoundSrvA**

8.104.1 Function Documentation

8.104.1.1 int main (int *argc*, char * *argv*[ ])

Definition at line 42 of file SoundSrvA.cc.

References clSoundSrvA::Main(), and SoundSrvA.

8.104.1.2 void SigHandler (int *iSigNum*)

Definition at line 48 of file SoundSrvA.cc.

References clLogFile::Add(), clSoundSrvA::Log, SoundSrvA, and clSoundSrv-A::Stop().

8.104.1.3 void* WrapAudioInThread (void * *vpParam*)

Definition at line 71 of file SoundSrvA.cc.

References clSoundSrvA::AudioInThread(), and SoundSrvA.

8.104.1.4 void* WrapServeClientThread (void * *vpParam*)

Definition at line 77 of file SoundSrvA.cc.

References clSoundSrvA::ServeClientThread(), and SoundSrvA.

8.104.2 Variable Documentation

8.104.2.1 bool bDaemon = false `[static]`

Definition at line 38 of file SoundSrvA.cc.

Referenced by clSoundSrvA::Main().

8.104.2.2 clSoundSrvA SoundSrvA

Definition at line 39 of file SoundSrvA.cc.

Referenced by main(), SigHandler(), WrapAudioInThread(), and WrapServe-ClientThread().

8.105 SoundSrvA.hh File Reference

`#include <Alloc.hh>`

`#include <Condition.hh>`

`#include <Mutex.hh>`

`#include <dsp/DSPOp.hh>`

`#include "Config.h"`

`#include "AudioA2.hh"`

`#include "CfgFile.hh"`

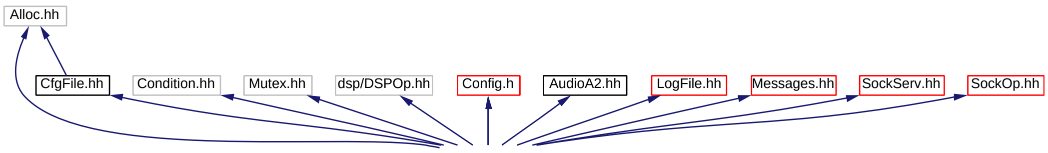
`#include "LogFile.hh"`

`#include "Messages.hh"`

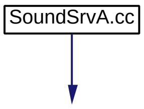
`#include "SockServ.hh"`

`#include "SockOp.hh"`

Include dependency graph for SoundSrvA.hh:



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



Compounds

- class **clSoundSrvA**

  SoundServer for ALSA.

Defines

- #define **SSA_LOGENTRY_SIZE** 256

8.105.1 Define Documentation

8.105.1.1 #define SSA_LOGENTRY_SIZE 256

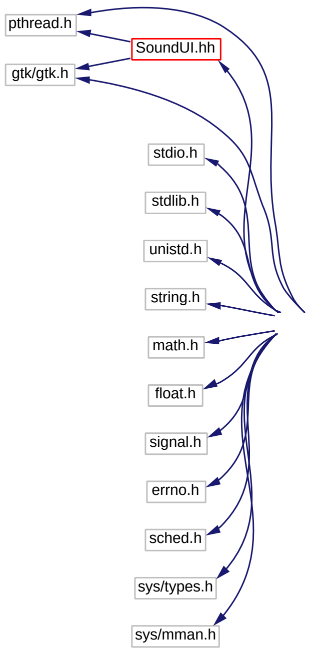
Definition at line 50 of file SoundSrvA.hh.

Referenced by clSoundSrvA::AudioInThread(), and clSoundSrvA::ServeClient-Thread().

8.106 SoundUI.cc File Reference

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <math.h>
#include <float.h>
#include <signal.h>
#include <errno.h>
#include <sched.h>
#include <sys/types.h>
#include <sys/mman.h>
#include <gtk/gtk.h>
#include "SoundUI.hh"
```

Include dependency graph for SoundUI.cc:



Functions

- int **main** (int argc, char *argv[ ])
- gboolean **WrapOnDeleteEvent** (GtkWidget *gwSender, GdkEvent *geEvent, gpointer gpData)
- void **WrapOnClickedEvent** (GtkButton *gbButton, gpointer gpData)
- gint **WrapOnTimeoutEvent** (gpointer gpData)
- void **WrapOnToggledEvent** (GtkToggleButton *gtbToggleButton, gpointer gpData)
- void **WrapOnValueChangedEvent** (GtkAdjustment *gaAdjustment, gpointer gpData)
- void **WrapOnApplyCurveClicked** (GtkButton *gbButton, gpointer gpData)
- void **WrapOnMotionCurve** (GtkWidget *gwSender, GdkEventMotion *gemEvent, gpointer gpData)
- void * **WrapSoundOutThread** (void *vpData)
- void * **WrapSoundInThread** (void *vpData)

Variables

- const char * **cpWindowTxt** = "Sound UI"
- const char * **cpChannelPfx** = "Channel"
- const char * **cpLServerTxt** = "Server"
- const char * **cpLChannelTxt** = "Channel"
- const char * **cpBConnectTxt** = "Connect"
- const char * **cpLInputLevelTxt** = "Average peak input level"
- const char * **cpLEqTxt** = "Eq"
- const char * **cpLOutputLevelTxt** = "Level"
- const char * **cpBApplyCurveTxt** = "Apply curve"
- **clSoundUI** * **SoundUI**

8.106.1 Function Documentation

8.106.1.1 int main (int *argc*, char * *argv*[ ])

Definition at line 56 of file SoundUI.cc.

References clSoundUI::Exec(), and SoundUI.

8.106.1.2 gboolean WrapOnDeleteEvent (GtkWidget * *gwSender*, GdkEvent * *geEvent*, gpointer *gpData*)

Definition at line 72 of file SoundUI.cc.

References clSoundUI::OnDeleteEvent(), and SoundUI.

Referenced by clSoundUI::ConnectSignals().

8.106.1.3 void WrapOnClickedEvent (GtkButton * *gbButton*, gpointer *gpData*)

Definition at line 79 of file SoundUI.cc.

References clSoundUI::OnClickedEvent(), and SoundUI.

Referenced by clSoundUI::ConnectSignals().

8.106.1.4 gint WrapOnTimeoutEvent (gpointer *gpData*)

Definition at line 85 of file SoundUI.cc.

References clSoundUI::OnTimeoutEvent(), and SoundUI.

Referenced by clSoundUI::Exec().

8.106.1.5 void WrapOnToggledEvent (GtkToggleButton * *gtbToggleButton*, gpointer *gpData*)

Definition at line 91 of file SoundUI.cc.

References clSoundUI::OnToggledEvent(), and SoundUI.

Referenced by clSoundUI::ConnectSignals().

8.106.1.6 void WrapOnValueChangedEvent (GtkAdjustment * *gaAdjustment*, gpointer *gpData*)

Definition at line 97 of file SoundUI.cc.

References clSoundUI::OnValueChangedEvent(), and SoundUI.

Referenced by clSoundUI::ConnectSignals().

8.106.1.7 void WrapOnApplyCurveClicked (GtkButton * *gbButton*, gpointer *gpData*)

Definition at line 103 of file SoundUI.cc.

References clSoundUI::OnApplyCurveClicked(), and SoundUI.

Referenced by clSoundUI::ConnectSignals().

8.106.1.8 void WrapOnMotionCurve (GtkWidget * *gwSender*, GdkEventMotion * *gemEvent*, gpointer *gpData*)

Definition at line 109 of file SoundUI.cc.

References clSoundUI::OnMotionCurve(), and SoundUI.

Referenced by clSoundUI::ConnectSignals().

8.106.1.9 void* WrapSoundOutThread (void * *vpData*)

Definition at line 116 of file SoundUI.cc.

References clSoundUI::SoundOutThread(), and SoundUI.

Referenced by clSoundUI::Exec().

8.106.1.10 void* WrapSoundInThread (void * *vpData*)

Definition at line 122 of file SoundUI.cc.

References clSoundUI::SoundInThread(), and SoundUI.

8.106.2 Variable Documentation

8.106.2.1 const char* cpWindowTxt = "Sound UI" [static]

Definition at line 40 of file SoundUI.cc.

Referenced by clSoundUI::BuildGUI(), and clSoundUI::clSoundUI().

8.106.2.2 const char* cpChannelPfx = "Channel" [static]

Definition at line 41 of file SoundUI.cc.

Referenced by clSoundChGUI::clSoundChGUI().

8.106.2.3 const char* cpLServerTxt = "Server" [static]

Definition at line 43 of file SoundUI.cc.

Referenced by clSoundChGUI::clSoundChGUI().

8.106.2.4 const char* cpLChannelTxt = "Channel" [static]

Definition at line 44 of file SoundUI.cc.

Referenced by clSoundChGUI::clSoundChGUI().

8.106.2.5 const char* cpBConnectTxt = "Connect" [static]

Definition at line 45 of file SoundUI.cc.

Referenced by clSoundChGUI::clSoundChGUI().

8.106.2.6 const char* cpLInputLevelTxt = "Average peak input level" [static]

Definition at line 47 of file SoundUI.cc.

Referenced by clSoundChGUI::clSoundChGUI().

8.106.2.7 const char* cpLEqTxt = "Eq" [static]

Definition at line 48 of file SoundUI.cc.

Referenced by clSoundChGUI::clSoundChGUI().

8.106.2.8 const char* cpLOutputLevelTxt = "Level" [static]

Definition at line 50 of file SoundUI.cc.

Referenced by clSoundChGUI::clSoundChGUI().

8.106.2.9 const char* cpBApplyCurveTxt = "Apply curve" [static]

Definition at line 51 of file SoundUI.cc.

Referenced by clSoundChGUI::clSoundChGUI().

8.106.2.10 clSoundUI* SoundUI

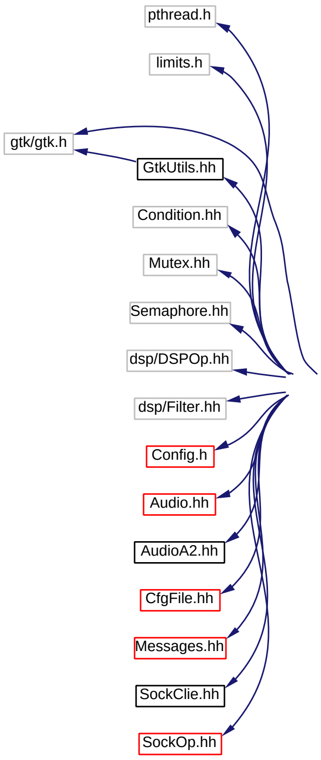
Definition at line 53 of file SoundUI.cc.

Referenced by main(), WrapOnApplyCurveClicked(), WrapOnClickedEvent(), WrapOnDeleteEvent(), WrapOnMotionCurve(), WrapOnTimeoutEvent(), WrapOnToggledEvent(), WrapOnValueChangedEvent(), WrapSoundInThread(), and WrapSoundOutThread().

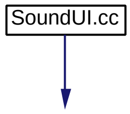
8.107 SoundUI.hh File Reference

```
#include <pthread.h>
#include <limits.h>
#include <gtk/gtk.h>
#include <Condition.hh>
#include <Mutex.hh>
#include <Semaphore.hh>
#include <dsp/DSPOp.hh>
#include <dsp/Filter.hh>
#include "Config.h"
#include "Audio.hh"
#include "AudioA2.hh"
#include "CfgFile.hh"
#include "GtkUtils.hh"
#include "Messages.hh"
#include "SockClie.hh"
#include "SockOp.hh"
```

Include dependency graph for SoundUI.hh:



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



Compounds

- class **clSoundChGUI**

  Per channel GUI for **clSoundUI**(p. 502).

- class **clSoundUI**

  Sound user interface.

Defines

- #define **SUI_VER_MAJ** GLOBAL_VERSMAJ
- #define **SUI_VER_MIN** GLOBAL_VERSMIN

- #define **SUI_VER_PL** GLOBAL_VERSPL
- #define **SUI_PADDING** 8
- #define **SUI_CONV_LEN** 256
- #define **SUI_SERV_MAXLEN** 255
- #define **SUI_CH_VALUE** 1.0
- #define **SUI_CH_LOWER** 1.0
- #define **SUI_CH_HIGHER** 16.0
- #define **SUI_IN_VALUE** 0.0
- #define **SUI_IN_LOWER** -72.0
- #define **SUI_IN_HIGHER** 0.0
- #define **SUI_OUT_VALUE** 0.0
- #define **SUI_OUT_LOWER** -72.0
- #define **SUI_OUT_HIGHER** 0.0
- #define **SUI_OUT_RANGE** 12.0
- #define **SUI_OUT_STEP** 0.1
- #define **SUI_EQ_RANGE** 12.0
- #define **SUI_EQ_STEP** 0.1
- #define **SUI_EQ_MAXOCTS** 15
- #define **SUI_FILTER_WINDOW** true

8.107.1 Define Documentation

8.107.1.1 #define SUI_VER_MAJ GLOBAL_VERSMAJ

Definition at line 55 of file SoundUI.hh.

Referenced by clSoundUI::clSoundUI().

8.107.1.2 #define SUI_VER_MIN GLOBAL_VERSMIN

Definition at line 56 of file SoundUI.hh.

Referenced by clSoundUI::clSoundUI().

8.107.1.3 #define SUI_VER_PL GLOBAL_VERSPL

Definition at line 57 of file SoundUI.hh.

Referenced by clSoundUI::clSoundUI().

8.107.1.4 #define SUI_PADDING 8

Definition at line 58 of file SoundUI.hh.

Referenced by clSoundUI::BuildGUI(), and clSoundChGUI::clSoundChGUI().

8.107.1.5 #define SUI_CONV_LEN 256

Definition at line 59 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI(), and clSoundUI::OnMotionCurve().

8.107.1.6 #define SUI_SERV_MAXLEN 255

Definition at line 60 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI(), clSoundUI::OnClickedEvent(), and clSoundUI::ParseServerStr().

8.107.1.7 #define SUI_CH_VALUE 1.0

Definition at line 61 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.8 #define SUI_CH_LOWER 1.0

Definition at line 62 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.9 #define SUI_CH_HIGHER 16.0

Definition at line 63 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.10 #define SUI_IN_VALUE 0.0

Definition at line 64 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.11 #define SUI_IN_LOWER -72.0

Definition at line 65 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.12 #define SUI_IN_HIGHER 0.0

Definition at line 66 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.13 #define SUI_OUT_VALUE 0.0

Definition at line 67 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.14 #define SUI_OUT_LOWER -72.0

Definition at line 68 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.15 #define SUI_OUT_HIGHER 0.0

Definition at line 69 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.16 #define SUI_OUT_RANGE 12.0

Definition at line 70 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.17 #define SUI_OUT_STEP 0.1

Definition at line 71 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.18 #define SUI_EQ_RANGE 12.0

Definition at line 72 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI(), and clSoundUI::OnMotionCurve().

8.107.1.19 #define SUI_EQ_STEP 0.1

Definition at line 73 of file SoundUI.hh.

Referenced by clSoundChGUI::clSoundChGUI().

8.107.1.20 #define SUI_EQ_MAXOCTS 15

Definition at line 74 of file SoundUI.hh.

Referenced by clSoundUI::Exec().

8.107.1.21 #define SUI_FILTER_WINDOW true

Definition at line 75 of file SoundUI.hh.

Referenced by clSoundUI::Exec().

8.108 SpectDir.cc File Reference

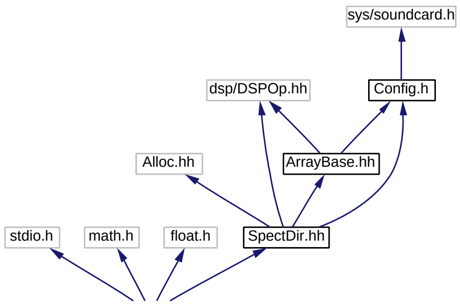
`#include <stdio.h>`

`#include <math.h>`

`#include <float.h>`

`#include "SpectDir.hh"`

Include dependency graph for SpectDir.cc:



8.109 SpectDir.hh File Reference

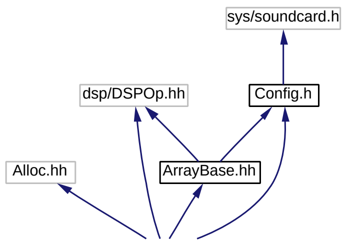
`#include <Alloc.hh>`

`#include <dsp/DSPOp.hh>`

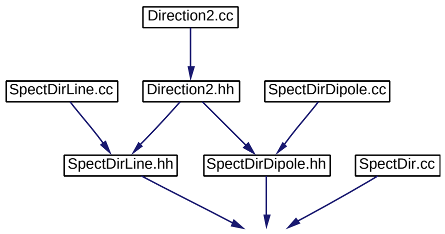
`#include "Config.h"`

`#include "ArrayBase.hh"`

Include dependency graph for SpectDir.hh:



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



Compounds

- struct **_stSpectDirRN**

  SpectDir: Noise removal parameters.

- class **clSpectDir**

  Direction calculation from spectrum.

Typedefs

- typedef **_stSpectDirRN stSpectDirRN**

  SpectDir: Noise removal parameters.

- typedef **_stSpectDirRN** ∗ **stpSpectDirRN**
  SpectDir: Noise removal parameters.

8.109.1 Typedef Documentation

8.109.1.1 typedef struct _stSpectDirRN stSpectDirRN

SpectDir: Noise removal parameters.

8.109.1.2 typedef struct _stSpectDirRN ∗ stpSpectDirRN

SpectDir: Noise removal parameters.

8.110 SpectDir2.cc File Reference

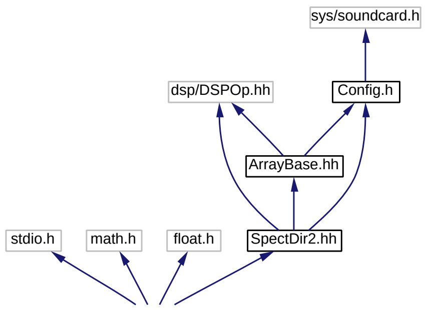
`#include <stdio.h>`

`#include <math.h>`

`#include <float.h>`

`#include "SpectDir2.hh"`

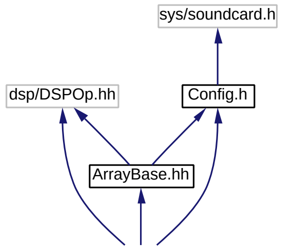
Include dependency graph for SpectDir2.cc:



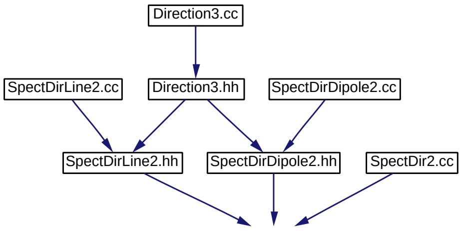
8.111 SpectDir2.hh File Reference

```
#include <dsp/DSPOp.hh>
#include "Config.h"
#include "ArrayBase.hh"
```

Include dependency graph for SpectDir2.hh:



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



Compounds

- struct **_stSpectDir2RN**
- class **clSpectDir2**

    Direction calculation from spectrum.

Typedefs

- typedef **_stSpectDir2RN stSpectDir2RN**
- typedef **_stSpectDir2RN** ∗ **stpSpectDir2RN**

8.111.1 Typedef Documentation

8.111.1.1 typedef struct _stSpectDir2RN stSpectDir2RN

8.111.1.2 typedef struct _stSpectDir2RN * stpSpectDir2RN

8.112 SpectDirDipole.cc File Reference

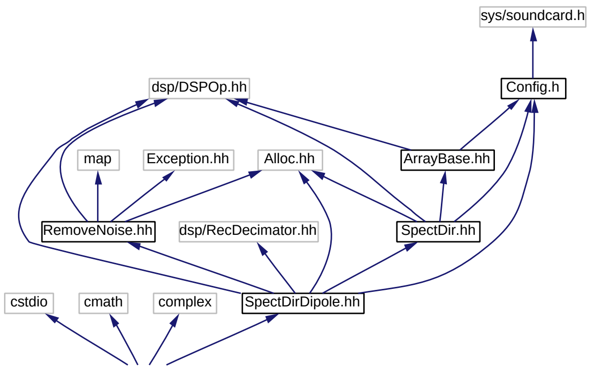
`#include <cstdio>`

`#include <cmath>`

`#include <complex>`

`#include "SpectDirDipole.hh"`

Include dependency graph for SpectDirDipole.cc:



8.113 SpectDirDipole.hh File Reference

`#include <Alloc.hh>`

`#include <dsp/DSPOp.hh>`

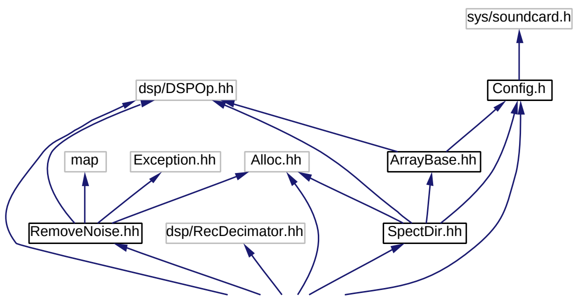
`#include <dsp/RecDecimator.hh>`

`#include "Config.h"`

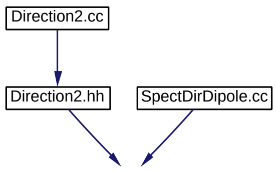
`#include "RemoveNoise.hh"`

`#include "SpectDir.hh"`

Include dependency graph for SpectDirDipole.hh:



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



Compounds

- class **clSpectDirDipole**

  Spectrum based direction finding for a dipole array.

Enumerations

- enum { **SDD_SCALE_LIN** = 0, **SDD_SCALE_LOG** = 1 }

  Scaling type.

- enum {
  SDD_BNER_NONE = 0, **SDD_BNER_TPSW** = 1, **SDD_BNER_-OTA** = 2, **SDD_BNER_DIFF** = 3,
  SDD_BNER_IDIFF = 4 }

  Background noise estimation and removal type.

8.113.1 Enumeration Type Documentation

8.113.1.1 anonymous enum

Scaling type.

Enumeration values:

SDD_SCALE_LIN Linear scaling.

SDD_SCALE_LOG Logarithmic (dB) scaling.

Definition at line 41 of file SpectDirDipole.hh.

8.113.1.2 anonymous enum

Background noise estimation and removal type.

Enumeration values:

SDD_BNER_NONE No noise removal.

SDD_BNER_TPSW Two-Pass Split-Window.

SDD_BNER_OTA Order-Truncate-Average.

SDD_BNER_DIFF Differential.

SDD_BNER_IDIFF Inverse differential.

Definition at line 50 of file SpectDirDipole.hh.

8.114 SpectDirDipole2.cc File Reference

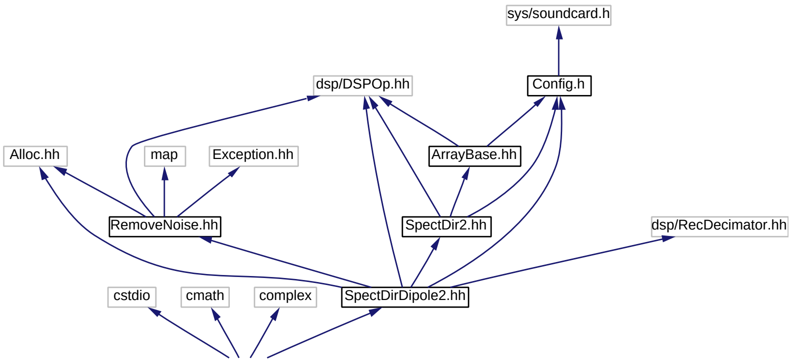
`#include <cstdio>`

`#include <cmath>`

`#include <complex>`

`#include "SpectDirDipole2.hh"`

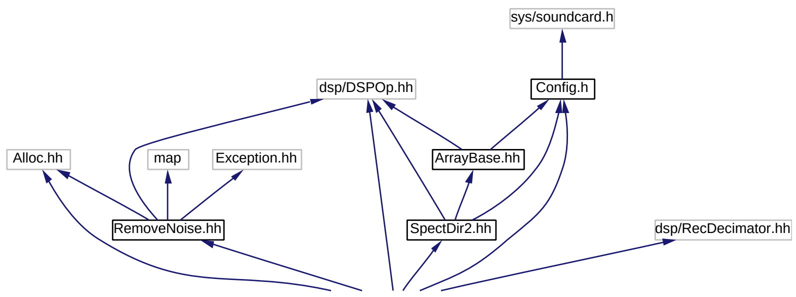
Include dependency graph for SpectDirDipole2.cc:



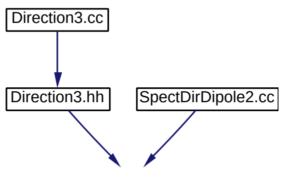
8.115 SpectDirDipole2.hh File Reference

```
#include <Alloc.hh>
#include <dsp/DSPOp.hh>
#include <dsp/RecDecimator.hh>
#include "Config.h"
#include "RemoveNoise.hh"
#include "SpectDir2.hh"
```

Include dependency graph for SpectDirDipole2.hh:



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



Compounds

- class **clSpectDirDipole2**

  Spectrum based direction finding for dipole array.

Enumerations

- enum { **SDD2_SCALE_LIN** = 0, **SDD2_SCALE_LOG** = 1 }
- enum {

  SDD2_BNER_NONE = 0, **SDD2_BNER_TPSW** = 1, **SDD2_BNER_OTA** = 2, **SDD2_BNER_DIFF** = 3,

SDD2_BNER_IDIFF = 4 }

8.115.1 Enumeration Type Documentation

8.115.1.1 anonymous enum

Enumeration values:

SDD2_SCALE_LIN

SDD2_SCALE_LOG

Definition at line 38 of file SpectDirDipole2.hh.

8.115.1.2 anonymous enum

Enumeration values:

SDD2_BNER_NONE

SDD2_BNER_TPSW

SDD2_BNER_OTA

SDD2_BNER_DIFF

SDD2_BNER_IDIFF

Definition at line 44 of file SpectDirDipole2.hh.

8.116 SpectDirLine.cc File Reference

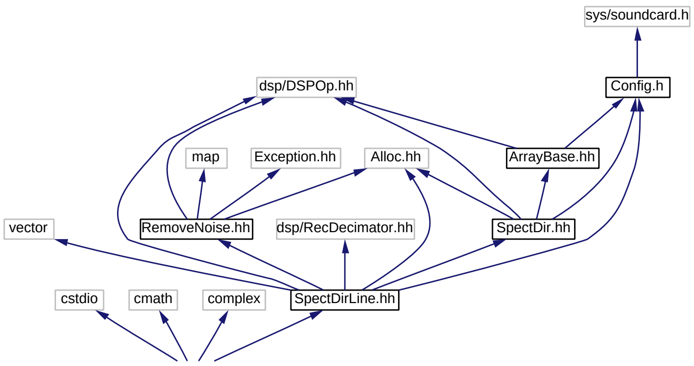
`#include <cstdio>`

`#include <cmath>`

`#include <complex>`

`#include "SpectDirLine.hh"`

Include dependency graph for SpectDirLine.cc:



8.117 SpectDirLine.hh File Reference

`#include <vector>`

`#include <Alloc.hh>`

`#include <dsp/DSPOp.hh>`

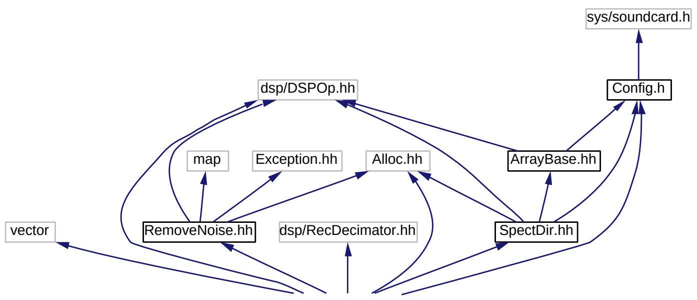
`#include <dsp/RecDecimator.hh>`

`#include "Config.h"`

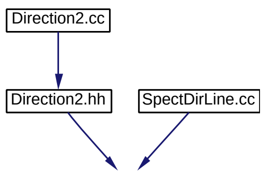
`#include "RemoveNoise.hh"`

`#include "SpectDir.hh"`

Include dependency graph for SpectDirLine.hh:



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



Compounds

- class **clSpectDirLine**

  Spectrum based direction finding for a line array.

Enumerations

- enum { **SDL_SCALE_LIN** = 0, **SDL_SCALE_LOG** = 1 }

  Scaling type.

- enum {
  SDL_BNER_NONE = 0, **SDL_BNER_TPSW** = 1, **SDL_BNER_-OTA** = 2, **SDL_BNER_DIFF** = 3,
  SDL_BNER_IDIFF = 4 }

  Background noise estimation and removal type.

8.117.1 Enumeration Type Documentation

8.117.1.1 anonymous enum

Scaling type.

Enumeration values:

SDL_SCALE_LIN Linear scaling.

SDL_SCALE_LOG Logarithmic (dB) scaling.

Definition at line 43 of file SpectDirLine.hh.

8.117.1.2 anonymous enum

Background noise estimation and removal type.

Enumeration values:

SDL_BNER_NONE No noise removal.

SDL_BNER_TPSW Two-Pass Split-Window.

SDL_BNER_OTA Order-Truncate-Average.

SDL_BNER_DIFF Differential.

SDL_BNER_IDIFF Inverse differential.

Definition at line 52 of file SpectDirLine.hh.

8.118 SpectDirLine2.cc File Reference

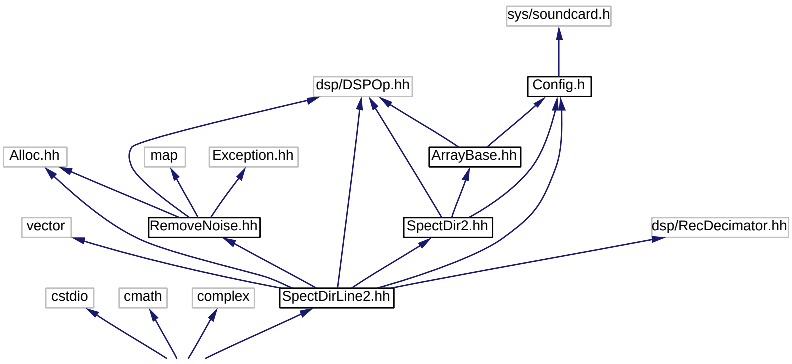
`#include <cstdio>`

`#include <cmath>`

`#include <complex>`

`#include "SpectDirLine2.hh"`

Include dependency graph for SpectDirLine2.cc:



8.119 SpectDirLine2.hh File Reference

`#include <vector>`

`#include <Alloc.hh>`

`#include <dsp/DSPOp.hh>`

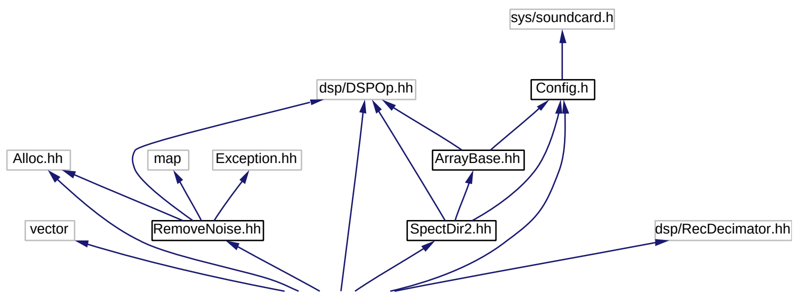
`#include <dsp/RecDecimator.hh>`

`#include "Config.h"`

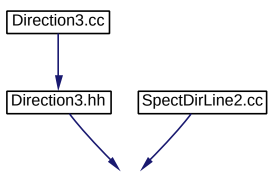
`#include "RemoveNoise.hh"`

`#include "SpectDir2.hh"`

Include dependency graph for SpectDirLine2.hh:



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



Compounds

- class **clSpectDirLine2**

  Spectrum based direction finding for line array.

Enumerations

- enum { **SDL2_SCALE_LIN** = 0, **SDL2_SCALE_LOG** = 1 }

- enum {
  SDL2_BNER_NONE = 0, **SDL2_BNER_TPSW** = 1, **SDL2_BNER_OTA** = 2, **SDL2_BNER_DIFF** = 3,
  SDL2_BNER_IDIFF = 4 }

8.119.1 Enumeration Type Documentation

8.119.1.1 anonymous enum

Enumeration values:
- **SDL2_SCALE_LIN**
- **SDL2_SCALE_LOG**

Definition at line 40 of file SpectDirLine2.hh.

8.119.1.2 anonymous enum

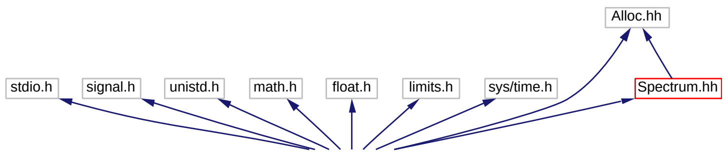
Enumeration values:
- **SDL2_BNER_NONE**
- **SDL2_BNER_TPSW**
- **SDL2_BNER_OTA**
- **SDL2_BNER_DIFF**
- **SDL2_BNER_IDIFF**

Definition at line 46 of file SpectDirLine2.hh.

8.120 Spectrum.cc File Reference

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <math.h>
#include <float.h>
#include <limits.h>
#include <sys/time.h>
#include <Alloc.hh>
#include "Spectrum.hh"
```

Include dependency graph for Spectrum.cc:



Defines

- #define **SPECT_NWIN** 13

Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char ∗argv[])

Variables

- bool **bDebug**
- bool **bDaemon**
- **clSpectrum** ∗ **Spectrum**
- const char ∗ **cpaWindowFuncs** []

8.120.1 Define Documentation

8.120.1.1 #define SPECT_NWIN 13

Definition at line 40 of file Spectrum.cc.

Referenced by clSpectrum::Init().

8.120.2 Function Documentation

8.120.2.1 void SigHandler (int *iSigNum*)

Definition at line 57 of file Spectrum.cc.

References bDebug.

8.120.2.2 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 73 of file Spectrum.cc.

References bDaemon, bDebug, clSpectrum::Main(), SigHandler(), and Spectrum.

8.120.3 Variable Documentation

8.120.3.1 bool bDebug `[static]`

Definition at line 36 of file Spectrum.cc.

Referenced by clSpectrum::CreateFilter(), clSpectrum::GetData(), clSpectrum::Init(), clSpectrum::Main(), main(), clSpectrum::ProcessLoop(), and SigHandler().

8.120.3.2 bool bDaemon `[static]`

Definition at line 37 of file Spectrum.cc.

Referenced by main().

8.120.3.3 clSpectrum∗ Spectrum

Definition at line 38 of file Spectrum.cc.

Referenced by main().

8.120.3.4 const char∗ cpaWindowFuncs[ ] `[static]`

Initial value:

```
{
    "Rectangle",
    "Bartlett",
    "Blackman",
    "Blackman-Harris",
    "Cosine tapered",
```

```
"Exponential",
"Flat-top",
"Generic cosine",
"Hamming",
"Hanning",
"Kaiser",
"Kaiser-Bessel",
"Tukey" }
```

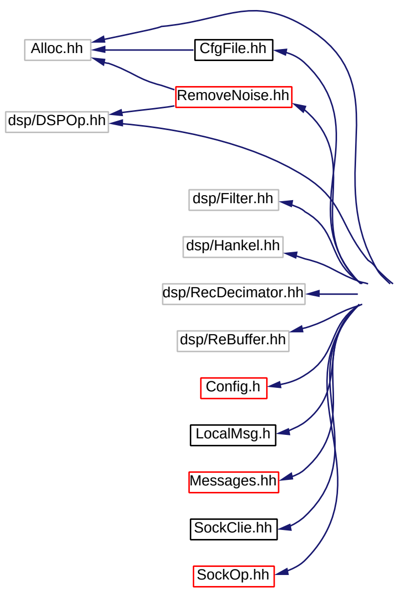
Definition at line 41 of file Spectrum.cc.

Referenced by clSpectrum::Init().

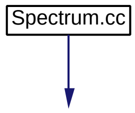
8.121 Spectrum.hh File Reference

```
#include <Alloc.hh>
#include <dsp/DSPOp.hh>
#include <dsp/Filter.hh>
#include <dsp/Hankel.hh>
#include <dsp/RecDecimator.hh>
#include <dsp/ReBuffer.hh>
#include "Config.h"
#include "LocalMsg.h"
#include "CfgFile.hh"
#include "Messages.hh"
#include "RemoveNoise.hh"
#include "SockClie.hh"
#include "SockOp.hh"
```

Include dependency graph for Spectrum.hh:



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



Compounds

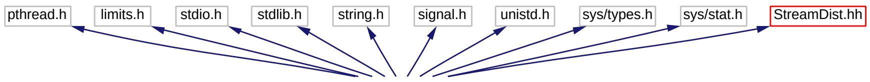
- class **clSpectrum**

  Spectrum server.

8.122 StreamDist.cc File Reference

```
#include <pthread.h>
#include <limits.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include "StreamDist.hh"
```

Include dependency graph for StreamDist.cc:



Functions

- int **main** (int argc, char *argv[ ])
- void **SigHandler** (int iSigNum)
- void * **WrapAudioInThread** (void *vpParam)
- void * **WrapServeClientThread** (void *vpParam)

Variables

- bool **bDaemon**
- **clStreamDist StreamDist**

8.122.1 Function Documentation

8.122.1.1 int main (int *argc*, char * *argv*[ ])

Definition at line 40 of file StreamDist.cc.

References clStreamDist::Main(), and StreamDist.

8.122.1.2 void SigHandler (int *iSigNum*)

Definition at line 46 of file StreamDist.cc.

References clLogFile::Add(), clStreamDist::Log, clStreamDist::Stop(), and StreamDist.

8.122.1.3 void* WrapAudioInThread (void * *vpParam*)

Definition at line 70 of file StreamDist.cc.

References clStreamDist::AudioInThread(), and StreamDist.

8.122.1.4 void* WrapServeClientThread (void * *vpParam*)

Definition at line 76 of file StreamDist.cc.

References clStreamDist::ServeClientThread(), and StreamDist.

8.122.2 Variable Documentation

8.122.2.1 bool bDaemon `[static]`

Definition at line 36 of file StreamDist.cc.

Referenced by clStreamDist::Main().

8.122.2.2 clStreamDist StreamDist

Definition at line 37 of file StreamDist.cc.

Referenced by main(), SigHandler(), WrapAudioInThread(), and WrapServeClientThread().

8.123 StreamDist.hh File Reference

#include <Alloc.hh>

#include <Condition.hh>

#include <Mutex.hh>

#include "Config.h"

#include "LocalMsg.h"

#include "CfgFile.hh"

#include "LogFile.hh"

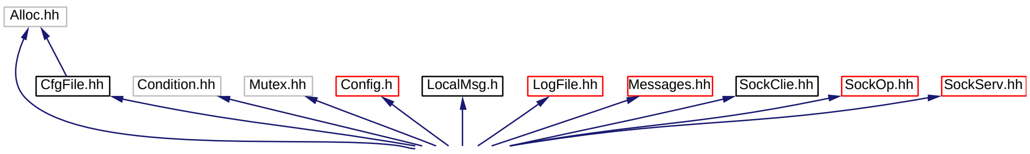
#include "Messages.hh"

#include "SockClie.hh"

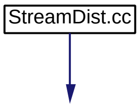
#include "SockOp.hh"

#include "SockServ.hh"

Include dependency graph for StreamDist.hh:



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



Compounds

- class **clStreamDist**

  Audio stream distributor.

Defines

- #define **SD_LOGENTRY_SIZE** 256
- #define **SD_HOSTADDR_MAX** 256

8.123.1 Define Documentation

8.123.1.1 #define SD_LOGENTRY_SIZE 256

Definition at line 46 of file StreamDist.hh.

Referenced by clStreamDist::AudioInThread().

8.123.1.2 #define SD_HOSTADDR_MAX 256

Definition at line 47 of file StreamDist.hh.

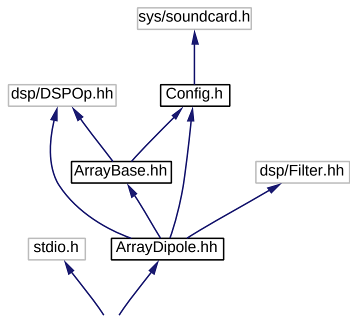
Referenced by clStreamDist::AudioInThread().

8.124 TestArray.cc File Reference

`#include <stdio.h>`

`#include "ArrayDipole.hh"`

Include dependency graph for TestArray.cc:



Functions

- int **main** (int argc, char ∗argv[ ])

8.124.1 Function Documentation

8.124.1.1 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 28 of file TestArray.cc.

References clArrayDipole::GetDelaySamples(), clArrayDipole::GetDelayTime(), clArrayDipole::Initialize(), and clArrayDipole::Uninitialize().

8.125 TestDir.cc File Reference

`#include <stdio.h>`

`#include <stdlib.h>`

`#include <string.h>`

`#include <math.h>`

`#include <float.h>`

`#include <signal.h>`

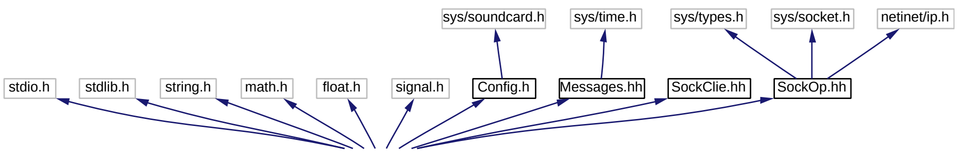
`#include "Config.h"`

`#include "Messages.hh"`

`#include "SockClie.hh"`

`#include "SockOp.hh"`

Include dependency graph for TestDir.cc:



Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char *argv[ ])

Variables

- bool **bRun** = true

8.125.1 Function Documentation

8.125.1.1 void SigHandler (int *iSigNum*)

Definition at line 39 of file TestDir.cc.

References bRun.

8.125.1.2 int main (int *argc*, char * *argv*[ ])

Definition at line 45 of file TestDir.cc.

References _stDirReq::bNormalize, bRun, clSockClie::Connect(), _stDirReq::fIntegrationTime, _stDirReq::fLeftDir, _stDirReq::fLowFreqLimit, _stDirReq::fRightDir, _stDirReq::fScalingExp, _stDirReq::fSoundSpeed, GDT, clDirMsg::GetResult(), GLOBAL_HEADER_LEN, _stDirReq::iAlgorithm, _stDirReq::iScaling, _stDirReq::lSectorCount, MSG_DIR_ALG_BEAM, MSG_DIR_SCAL_LIN, clSockOp::ReadN(), clDirMsg::SetRequest(), SigHandler(), SOp, and clSockOp::WriteN().

8.125.2 Variable Documentation

8.125.2.1 bool bRun = true

Definition at line 36 of file TestDir.cc.

8.126 TestLocate.cc File Reference

`#include <stdio.h>`

`#include <signal.h>`

`#include <ptc/ptc.h>`

`#include <Alloc.hh>`

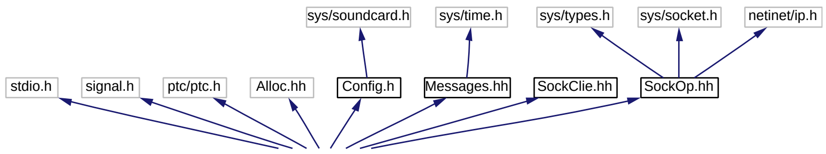
`#include "Config.h"`

`#include "Messages.hh"`

`#include "SockClie.hh"`

`#include "SockOp.hh"`

Include dependency graph for TestLocate.cc:



Functions

- Format **Fmt** (32, 0x00ff0000, 0x0000ff00, 0x000000ff)
- void **SigHandler** (int iSigNum)
- void **PlotData** (Surface &Sfc, GDT ∗fpData, int iWidth, int iHeight)
- int **main** (int argc, char ∗argv[ ])

Variables

- bool **bRun** = true
- Console **Con**

8.126.1 Function Documentation

8.126.1.1 Format Fmt (32, 0x00ff0000, 0x0000ff00, 0x000000ff)

Referenced by main().

8.126.1.2 void SigHandler (int *iSigNum*)

Definition at line 41 of file TestLocate.cc.

References bRun.

8.126.1.3 void PlotData (Surface & *Sfc*, GDT ∗ *fpData*, int *iWidth*, int *iHeight*)

Definition at line 47 of file TestLocate.cc.

Referenced by main().

8.126.1.4 int main (int *argc*, char ∗ *argv*[ ])

Definition at line 74 of file TestLocate.cc.

References bRun, Con, clSockClie::Connect(), Fmt(), GDT, clLocateMsg::GetHeader(), clLocateMsg::GetResult(), GLOBAL_HEADER_LEN, _stLocateHdr::iHeight, _stLocateHdr::iWidth, PlotData(), clSockOp::ReadN(), clSockOp::SetHandle(), SigHandler(), and SOp.

8.126.2 Variable Documentation

8.126.2.1 bool bRun = true

Definition at line 36 of file TestLocate.cc.

8.126.2.2 Console Con

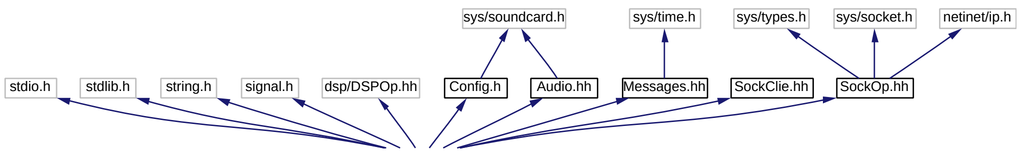
Definition at line 37 of file TestLocate.cc.

Referenced by main().

8.127 TestSnd.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <dsp/DSPOp.hh>
#include "Config.h"
#include "Audio.hh"
#include "Messages.hh"
#include "SockClie.hh"
#include "SockOp.hh"
```

Include dependency graph for TestSnd.cc:



Functions

- void **SigHandler** (int iSignal)
- int **main** (int argc, char *argv[])

Variables

- bool **bRun** = true

8.127.1 Function Documentation

8.127.1.1 void SigHandler (int *iSignal*)

Definition at line 40 of file TestSnd.cc.

References bRun.

8.127.1.2 int main (int *argc*, char * *argv*[])

Definition at line 54 of file TestSnd.cc.

References AUDIO_WRITE, bRun, clSockClie::Connect(), _stSoundStart::dSampleRate, GDT, clSoundMsg::GetData(), clAudio::GetError(), clAudio::GetOutBufInfo(), clSoundMsg::GetStart(), GLOBAL_HEADER_LEN, _stSoundStart::iChannels, clSockOp::Read(), clSockOp::Shutdown(), SigHandler(), and clAudio::Write().

8.127.2 Variable Documentation

8.127.2.1 bool bRun = true

Definition at line 37 of file TestSnd.cc.

8.128 UIServ.cc File Reference

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <signal.h>

#include <fcntl.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <sys/wait.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include "Config.h"

#include "UIServ.hh"

#include "SockOp.hh"

Include dependency graph for UIServ.cc:

Functions

- void **SigHandler** (int iSigNum)
- int **main** (int argc, char *argv[])

Variables

- bool **bRun** = true
- **clUIServer** * **UIServer**
- **clSockOp** * **SOp**

8.128.1 Function Documentation

8.128.1.1 void SigHandler (int *iSigNum*)

Definition at line 46 of file UIServ.cc.

References bRun.

8.128.1.2 int main (int *argc*, char * *argv*[ ])

Definition at line 61 of file UIServ.cc.

References bRun, clSockOp::DisableNagle(), GLOBAL_HEADER_LEN, clUIServer::NoLogShutdown(), clSockOp::ReadN(), clSockOp::ReadSelect(), clSockOp::SetTypeOfService(), SigHandler(), SOp, UIS_MSG_TIMEOUT, UIS_VERSMAJ, UIS_VERSMIN, UIS_VERSPL, UIServer, and clUIServer::Wait().

8.128.2 Variable Documentation

8.128.2.1 bool bRun = true

Definition at line 41 of file UIServ.cc.

8.128.2.2 clUIServer* UIServer

Definition at line 42 of file UIServ.cc.

Referenced by main().

8.128.2.3 clSockOp* SOp

Definition at line 43 of file UIServ.cc.

Referenced by clStreamDist::AudioInThread(), main(), clXMMSOut::ServeClientThread(), clStreamDist::ServeClientThread(), clSoundSrvA::ServeClientThread(), clSoundSrv2::ServeClientThread(), clSoundProxy::ServeClientThread(), clLocate::ServeClientThread(), clFileSrv::ServeClientThread(), clComediSrv::ServeClientThread(), clBeamSrv2Master::ServeClientThread(), clBeamSrvMaster::ServeClientThread(), clSoundUI::SoundInThread(), clSoundProxy::SoundInThread(), and clUIServer::Wait().

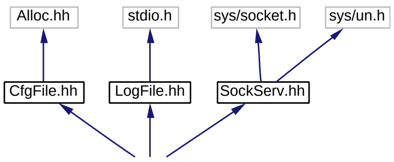
8.129 UIServ.hh File Reference

`#include "CfgFile.hh"`

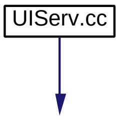
`#include "LogFile.hh"`

`#include "SockServ.hh"`

Include dependency graph for UIServ.hh:



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



Compounds

- class **clUIServer**

  User interface server.

Defines

- #define **UIS_CONV_BUF_SIZE** 255

8.129.1 Define Documentation

8.129.1.1 #define UIS_CONV_BUF_SIZE 255

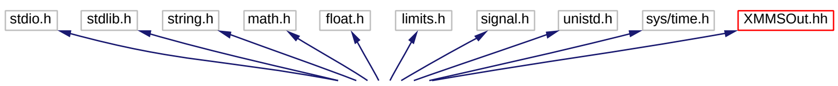
Definition at line 31 of file UIServ.hh.

Referenced by clUIServer::clUIServer(), and clUIServer::Wait().

8.130 XMMSOut.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <float.h>
#include <limits.h>
#include <signal.h>
#include <unistd.h>
#include <sys/time.h>
#include "XMMSOut.hh"
```

Include dependency graph for XMMSOut.cc:



Functions

- OutputPlugin * **get_oplugin_info** (void)
- void **hasas_init** (void)
- void **hasas_about** (void)
- void **hasas_configure** (void)
- void **hasas_get_volume** (int *, int *)
- void **hasas_set_volume** (int, int)
- int **hasas_open_audio** (AFormat, int, int)
- void **hasas_write_audio** (void *, int)
- void **hasas_close_audio** (void)
- void **hasas_flush** (int)
- void **hasas_pause** (short)
- int **hasas_buffer_free** (void)
- int **hasas_buffer_playing** (void)
- int **hasas_output_time** (void)
- int **hasas_written_time** (void)
- clDynThreads< **clXMMSOut** > **XMMSOutThreads** (**XMMSOut**)
- void **WrapOnAboutButton** (GtkButton *gbSender, gpointer gpData)
- void **WrapOnButtonClick** (GtkButton *gbSender, gpointer gpData)

Variables

- OutputPlugin **hasas_op**
- **clXMMSOut XMMSOut**
- const int **iDefBufferSize** = 65536
- const char * **cpDefLocalSocket** = "streamdist.socket"
- const char * **cpWinTitleTxt** = "HASAS streamdist settings"
- const char * **cpLBufSizeTxt** = "Buffer size"
- const char * **cpLLocalSocketTxt** = "Local socket"
- const char * **cpBOkTxt** = "OK"
- const char * **cpBCancelTxt** = "Cancel"

8.130.1 Function Documentation

8.130.1.1 OutputPlugin * get_oplugin_info (void)

Definition at line 91 of file XMMSOut.cc.

References GLOBAL_VERSMAJ, GLOBAL_VERSMIN, GLOBAL_VERSPL, hasas_op, and XMMSOUT_DESCRIPTION.

8.130.1.2 void hasas_init (void)

Definition at line 100 of file XMMSOut.cc.

References clXMMSOut::Init(), and XMMSOut.

8.130.1.3 void hasas_about (void)

Definition at line 106 of file XMMSOut.cc.

References clXMMSOut::About(), and XMMSOut.

8.130.1.4 void hasas_configure (void)

Definition at line 112 of file XMMSOut.cc.

References clXMMSOut::Configure(), and XMMSOut.

8.130.1.5 void hasas_get_volume (int *, int *)

8.130.1.6 void hasas_set_volume (int, int)

8.130.1.7 int hasas_open_audio (AFormat, int, int)

Definition at line 118 of file XMMSOut.cc.

References clXMMSOut::OpenAudio(), and XMMSOut.

8.130.1.8 void hasas_write_audio (void *, int)

Definition at line 124 of file XMMSOut.cc.

References clXMMSOut::WriteAudio(), and XMMSOut.

8.130.1.9 void hasas_close_audio (void)

Definition at line 130 of file XMMSOut.cc.

References clXMMSOut::CloseAudio(), and XMMSOut.

8.130.1.10 void hasas_flush (int)

Definition at line 136 of file XMMSOut.cc.

References clXMMSOut::Flush(), and XMMSOut.

8.130.1.11 void hasas_pause (short)

Definition at line 142 of file XMMSOut.cc.

References clXMMSOut::Pause(), and XMMSOut.

8.130.1.12 int hasas_buffer_free (void)

Definition at line 148 of file XMMSOut.cc.

References clXMMSOut::BufferFree(), and XMMSOut.

8.130.1.13 int hasas_buffer_playing (void)

Definition at line 154 of file XMMSOut.cc.

References clXMMSOut::BufferPlaying(), and XMMSOut.

8.130.1.14 int hasas_output_time (void)

Definition at line 160 of file XMMSOut.cc.

References clXMMSOut::OutputTime(), and XMMSOut.

8.130.1.15 int hasas_written_time (void)

Definition at line 166 of file XMMSOut.cc.

References clXMMSOut::WrittenTime(), and XMMSOut.

8.130.1.16 clDynThreads<clXMMSOut> XMMSOutThreads (XMMSOut)

Referenced by clXMMSOut::CloseAudio(), clXMMSOut::MainThread(), and clXMMSOut::OpenAudio().

8.130.1.17 void WrapOnAboutButton (GtkButton * *gbSender*, gpointer *gpData*)

Definition at line 172 of file XMMSOut.cc.

References clXMMSOut::OnAboutButton(), and XMMSOut.

Referenced by clXMMSOut::About().

8.130.1.18 void WrapOnButtonClick (GtkButton * *gbSender*, gpointer *gpData*)

Definition at line 178 of file XMMSOut.cc.

References clXMMSOut::OnButtonClick(), and XMMSOut.

Referenced by clXMMSOut::Configure().

8.130.2 Variable Documentation

8.130.2.1 OutputPlugin hasas_op

Initial value:

```
 {
    NULL,
    NULL,
    NULL,
    hasas_init,
    hasas_about,
    hasas_configure,
    NULL,
    NULL,
    hasas_open_audio,
    hasas_write_audio,
    hasas_close_audio,
    hasas_flush,
    hasas_pause,
    hasas_buffer_free,
    hasas_buffer_playing,
    hasas_output_time,
    hasas_written_time
}
```

Definition at line 56 of file XMMSOut.cc.

Referenced by get_oplugin_info().

8.130.2.2 clXMMSOut XMMSOut

Definition at line 75 of file XMMSOut.cc.

Referenced by hasas_about(), hasas_buffer_free(), hasas_buffer_playing(), hasas_close_audio(), hasas_configure(), hasas_flush(), hasas_init(), hasas_open_audio(), hasas_output_time(), hasas_pause(), hasas_write_audio(), hasas_written_time(), WrapOnAboutButton(), and WrapOnButtonClick().

8.130.2.3 const int iDefBufferSize = 65536 [static]

Definition at line 77 of file XMMSOut.cc.

Referenced by clXMMSOut::Init().

8.130.2.4 const char∗ cpDefLocalSocket = "streamdist.socket" [static]

Definition at line 78 of file XMMSOut.cc.

Referenced by clXMMSOut::Init().

8.130.2.5 const char∗ cpWinTitleTxt = "HASAS streamdist settings" [static]

Definition at line 80 of file XMMSOut.cc.

Referenced by clXMMSOut::Configure().

8.130.2.6 const char∗ cpLBufSizeTxt = "Buffer size" [static]

Definition at line 81 of file XMMSOut.cc.

Referenced by clXMMSOut::Configure().

8.130.2.7 const char∗ cpLLocalSocketTxt = "Local socket" [static]

Definition at line 82 of file XMMSOut.cc.

Referenced by clXMMSOut::Configure().

8.130.2.8 const char∗ cpBOkTxt = "OK" [static]

Definition at line 83 of file XMMSOut.cc.

Referenced by clXMMSOut::Configure().

8.130.2.9 const char* cpBCancelTxt = "Cancel" [static]

Definition at line 84 of file XMMSOut.cc.

Referenced by clXMMSOut::Configure().

8.131 XMMSOut.hh File Reference

`#include <gdk/gdk.h>`

`#include <gtk/gtk.h>`

`#include <xmms/configfile.h>`

`#include <xmms/plugin.h>`

`#include <xmms/util.h>`

`#include <Alloc.hh>`

`#include <Condition.hh>`

`#include <Mutex.hh>`

`#include <dsp/DSPOp.hh>`

`#include <dsp/ReBufferT.hh>`

`#include "Config.h"`

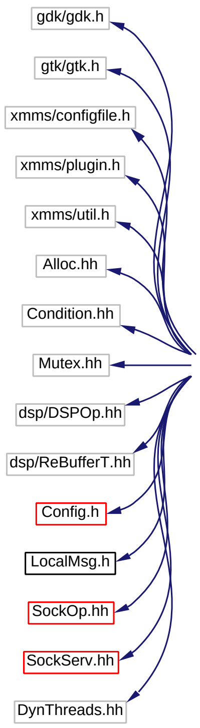
`#include "LocalMsg.h"`

`#include "SockOp.hh"`

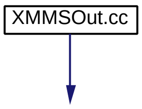
`#include "SockServ.hh"`

`#include "DynThreads.hh"`

Include dependency graph for XMMSOut.hh:



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



Compounds

- class **clXMMSOut**

Defines

- #define **XMMSOUT_DESCRIPTION** "HASAS streamdist"

8.131.1 Define Documentation

8.131.1.1 #define XMMSOUT_DESCRIPTION "HASAS streamdist"

Definition at line 45 of file XMMSOut.hh.

Referenced by clXMMSOut::About(), and get_oplugin_info().

Index