MOOSGenLib Reference Manual

Generated by Doxygen 1.4.6

Fri Sep 28 20:14:21 2007

Contents

Chapter 1

MOOSGenLib Hierarchical Index

1.1 MOOSGenLib Class Hierarchy

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

Chapter 2

MOOSGenLib Class Index

2.1 MOOSGenLib Class List

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

Chapter 3

MOOSGenLib File Index

3.1 MOOSGenLib File List

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

Chapter 4

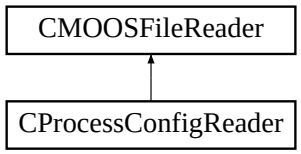
MOOSGenLib Class Documentation

4.1 CMOOSFileReader Class Reference

Base class for reading ascii files.

`#include <MOOSFileReader.h>`

Inheritance diagram for CMOOSFileReader::



Public Member Functions

- bool **IsOpen** ()
- bool **GoTo** (std::string sLine)
- bool **eof** ()
- bool **Reset** ()
- bool **GetValue** (std::string sName, std::string &sResult)
- bool **GetValue** (std::string sName, double &dfResult)
- bool **GetValue** (std::string sName, int &nResult)
- bool **GetValue** (std::string sName, float &fResult)
- bool **GetValue** (std::string sName, bool &bResult)
- bool **GetValue** (std::string sName, unsigned int &nResult)
- bool **SetFile** (const std::string &sFile)
- std::string **GetNextValidLine** ()

Static Public Member Functions

- static bool **GetTokenValPair** (std::string sLine, std::string &sTok, std::string &sVal, bool bPreserveWhiteSpace=false)

Protected Member Functions

- std::ifstream * **GetFile** ()

Static Protected Member Functions

- static bool **IsComment** (std::string &sLine)

Protected Attributes

- **CMOOSLock** * **m_pLock**
- std::string **m_sFileName**
- std::ifstream **m_File**
- THREAD2FILE_MAP **m_FileMap**

4.1.1 Detailed Description

Base class for reading ascii files.

4.1.2 Member Function Documentation

4.1.2.1 std::string CMOOSFileReader::GetNextValidLine ()

returns a string of teh next non comment line (and removs trailing comments)

4.1.2.2 bool CMOOSFileReader::GetTokenValPair (std::string *sLine*, std::string & *sTok*, std::string & *sVal*, bool *bPreserveWhiteSpace* = `false`) `[static]`

static helper which splits a line into token = value and by deafult removes white space

4.1.2.3 bool CMOOSFileReader::GetValue (std::string *sName*, unsigned int & *nResult*)

looks for a line "sName = Val" in whole file, fills in result with Val

4.1.2.4 bool CMOOSFileReader::GetValue (std::string *sName*, bool & *bResult*)

looks for a line "sName = Val" in whole file, fills in result with Val

4.1.2.5 bool CMOOSFileReader::GetValue (std::string *sName*, float & *fResult*)

looks for a line "sName = Val" in whole file, fills in result with Val

4.1.2.6 bool CMOOSFileReader::GetValue (std::string *sName*, int & *nResult*)

looks for a line "sName = Val" in whole file, fills in result with Val

4.1.2.7 bool CMOOSFileReader::GetValue (std::string *sName*, double & *dfResult*)

looks for a line "sName = Val" in whole file, fills in result with Val

4.1.2.8 bool CMOOSFileReader::GetValue (std::string *sName*, std::string & *sResult*)

looks for a line "sName = Val" in whole file, fills in result with Val

4.1.2.9 bool CMOOSFileReader::SetFile (const std::string & *sFile*)

tell the class what file to read

4.1.3 Member Data Documentation

4.1.3.1 THREAD2FILE_MAP CMOOSFileReader::m_FileMap [protected]

every thread get its own pointer to a stream

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

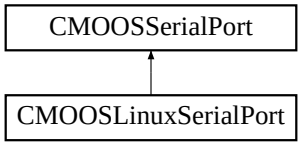
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSFileReader.h
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSFileReader.cpp

4.2 CMOOSLinuxSerialPort Class Reference

Implements linux aspects of **CMOOSSerialPort**(p. 15).

`#include <MOOSLinuxSerialPort.h>`

Inheritance diagram for CMOOSLinuxSerialPort::



Public Member Functions

- virtual bool **Close** ()
- **CMOOSLinuxSerialPort** ()
- virtual ~**CMOOSLinuxSerialPort** ()
- virtual bool **Create** (const char *pPortNum=DEFAULT_PORT, int nBaudRate=DEFAULT_BAUDRATE)
- int **Write** (char *Str, int nLen, double *pTime=NULL)
- virtual void **Break** ()
- virtual int **Flush** ()
- int **GetFD** ()

Protected Member Functions

- virtual int **GrabN** (char *pBuffer, int nRequired)

Protected Attributes

- int **m_nPortFD**
- termios **m_OldPortOptions**
- termios **m_PortOptions**

4.2.1 Detailed Description

Implements linux aspects of **CMOOSSerialPort**(p. 15).

4.2.2 Constructor & Destructor Documentation

4.2.2.1 CMOOSLinuxSerialPort::CMOOSLinuxSerialPort ()

constructor.

4.2.2.2 CMOOSLinuxSerialPort::~CMOOSLinuxSerialPort () [virtual]

Destructor.Reset the port option to what every they were before and close port

4.2.3 Member Function Documentation

4.2.3.1 void CMOOSLinuxSerialPort::Break () [virtual]

send break signal

Reimplemented from **CMOOSSerialPort** (p. 15).

4.2.3.2 bool CMOOSLinuxSerialPort::Create (const char * *pPortNum* = DEFAULT_PORT, int *nBaudRate* = DEFAULT_BAUDRATE) [virtual]

Create and set up the port

Implements **CMOOSSerialPort** (p. 15).

4.2.3.3 int CMOOSLinuxSerialPort::Flush (void) [virtual]

Call this method in order to free the Output Buffer of any characters that may not have been sent during our last write. We use the queue_selector TCOFLUSH.

See also:
http://www.mkssoftware.com/docs/man3/tcflush.3.asp

Reimplemented from **CMOOSSerialPort** (p. 15).

4.2.3.4 int CMOOSLinuxSerialPort::GetFD ()

returns the file descriptor

4.2.3.5 int CMOOSLinuxSerialPort::GrabN (char * *pBuffer*, int *nRequired*) [protected, virtual]

Just grab N characters NOW

Implements **CMOOSSerialPort** (p. 15).

4.2.3.6 int CMOOSLinuxSerialPort::Write (char * *Str*, int *nLen*, double * *pTime* = NULL) [virtual]

Write a string out of port

Implements **CMOOSSerialPort** (p. 15).

4.2.4 Member Data Documentation

4.2.4.1 int CMOOSLinuxSerialPort::m_nPortFD [protected]

FileDescriptor of Port

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

- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSLinuxSerialPort.h
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSLinuxSerialPort.cpp

4.3 CMOOSLock Class Reference

`#include <MOOSLock.h>`

Public Member Functions

- void **UnLock** ()
  call this to unlock
- void **Lock** ()
  call this to lock
- **CMOOSLock** (bool bInitial=true)

Protected Attributes

- pthread_mutex_t **m_hLock**
  posix mutex

4.3.1 Detailed Description

A very simple cross platform posix and win32 compatible mutex class

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

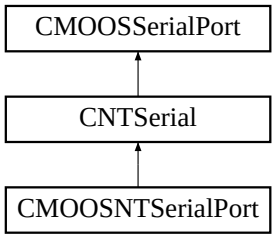
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSLock.h
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSLock.cpp

4.4 CMOOSNTSerialPort Class Reference

Implements windows specfialisations of MOOSSerialPort.

`#include <MOOSNTSerialPort.h>`

Inheritance diagram for CMOOSNTSerialPort::



Public Member Functions

- virtual void **Break** ()
- virtual bool **Create** (const char *pPortNum=DEFAULT_PORT, int nBaudRate=DEFAULT_BAUDRATE)
- bool **Close** (void)
- int **Write** (char *pData, int nLen, double *pTime=NULL)

Protected Member Functions

- virtual int **GrabN** (char *pBuffer, int nRequired)

4.4.1 Detailed Description

Implements windows specfialisations of MOOSSerialPort.

4.4.2 Member Function Documentation

4.4.2.1 void CMOOSNTSerialPort::Break () [virtual]

Send break signal

Reimplemented from **CMOOSSerialPort** (p. 15).

4.4.2.2 bool CMOOSNTSerialPort::Close (void) [virtual]

Close Port

Reimplemented from **CMOOSSerialPort** (p. 15).

4.4.2.3 bool CMOOSNTSerialPort::Create (const char * *pPortNum* = DEFAULT_PORT, int *nBaudRate* = DEFAULT_BAUDRATE) [virtual]

Create an open port

Implements **CMOOSSerialPort** (p. 15).

4.4.2.4 int CMOOSNTSerialPort::GrabN (char * *pBuffer*, int *nRequired*) [protected, virtual]

Grab N chars NOW

Implements **CMOOSSerialPort** (p. 15).

4.4.2.5 int CMOOSNTSerialPort::Write (char * *pData*, int *nLen*, double * *pTime* = NULL) [virtual]

Write nLen bytes out

Implements **CMOOSSerialPort** (p. 15).

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

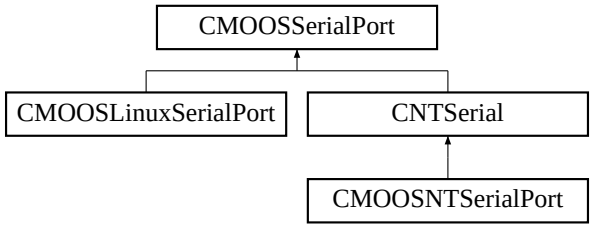
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSNTSerialPort.h
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSNTSerialPort.cpp

4.5 CMOOSSerialPort Class Reference

Cross Platform Serial Port Base Class.

`#include <MOOSSerialPort.h>`

Inheritance diagram for CMOOSSerialPort::



Public Types

- typedef std::list< CMOOSSerialTelegram > **TELEGRAM_LIST**

Public Member Functions

- std::string **GetPortName** ()
- virtual bool **Close** ()
- char **GetTermCharacter** ()
- void **SetTermCharacter** (char cTermChar)
- int **GetBaudRate** ()
- virtual int **Flush** ()
- bool **IsStreaming** ()
- bool **IsVerbose** ()
- bool **GetLatest** (std::string &sWhat, double &dfWhen)
- bool **GetEarliest** (std::string &sWhat, double &dfWhen)
- bool **CommsLoop** ()
- virtual bool **Configure** (STRING_LIST sParams)
- virtual bool **Create** (const char *pPortNum=DEFAULT_PORT, int nBaudRate=DEFAULT_BAUDRATE)=0
- virtual int **ReadNWithTimeOut** (char *pBuff, int nBufferLen, double Timeout=0.5, double *pTime=NULL)
- virtual int **Write** (char *Str, int nLen, double *pTime=NULL)=0
- bool **GetTelegram** (std::string &sTelegram, double dfTimeOut, double *pTime=NULL)
- void **SetIsCompleteReplyCallBack** (bool(*pfn)(char *pData, int nLen, int nRead))
- virtual void **Break** ()

Public Attributes

- TELEGRAM_LIST **m_InBox**
- TELEGRAM_LIST **m_OutBox**
- **CMOOSLock m_InBoxLock**
- **CMOOSLock m_OutBoxLock**
- **CMOOSLock m_PortLock**

Protected Types

- typedef pthread_t **THREAD_ID**

Protected Member Functions

- bool **StartThreads** ()
- virtual int **GrabN** (char *pBuffer, int nRequired)=0
- bool **IsCompleteReply** (char *pData, int nLen, int nRead)

Protected Attributes

- char **m_cTermCharacter**
- **THREAD_ID m_nCommsThreadID**
- bool **m_bStreaming**
- bool **m_bVerbose**
- bool(* **m_pfnUserIsCompleteReplyCallBack**)(char *pData, int nLen, int nRead)
- bool **m_bHandShaking**

  handware handshaking active flag

- std::string **m_sPort**

  port name

- int **m_nBaudRate**

  baudrate

- bool **m_bQuit**
- bool **m_bUseCsmExt**

  ARH 14/05/2005 For 500kBaud PCMCIA card.

Classes

- class **CMOOSSerialTelegram**

4.5.1 Detailed Description

Cross Platform Serial Port Base Class.

Provides cross platform functionality which is implemented in detail by the platform dependent derivatives

4.5.2 Member Typedef Documentation

4.5.2.1 typedef pthread_t CMOOSSerialPort::THREAD_ID [protected]

Win32 handle to IO thread

4.5.3 Member Function Documentation

4.5.3.1 void CMOOSSerialPort::SetTermCharacter (char *cTermChar*)

Sets the termination character for the serial port to watch out for when it constructs Telegrams for Streaming Devices.

4.5.4 Member Data Documentation

4.5.4.1 THREAD_ID CMOOSSerialPort::m_nCommsThreadID [protected]

ID of IO thread

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

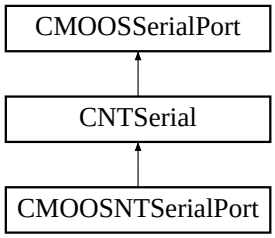
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/**MOOSSerialPort.h**
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSSerialPort.cpp

4.6 CNTSerial Class Reference

Middle Layer class for Windows Serial port (c) Ramon de Klein.

`#include <NTSerial.h>`

Inheritance diagram for CNTSerial::



Public Types

- enum **EEvent** {

  EEventNone = -1, **EEventBreak** = EV_BREAK, **EEventCTS** = EV_CTS, **EEventDSR** = EV_DSR,

  EEventError = EV_ERR, **EEventRing** = EV_RING, **EEventRLSD** = EV_RLSD, **EEventRecv** = EV_RXCHAR,

  EEventRcvEv = EV_RXFLAG, **EEventSend** = EV_TXEMPTY }
- enum **EBaudrate** {

  EBaudUnknown = -1, **EBaud110** = CBR_110, **EBaud300** = CBR_300, **EBaud600** = CBR_600,

  EBaud1200 = CBR_1200, **EBaud2400** = CBR_2400, **EBaud4800** = CBR_4800, **EBaud9600** = CBR_9600,

  EBaud14400 = CBR_14400, **EBaud19200** = CBR_19200, **EBaud38400** = CBR_38400, **EBaud56000** = CBR_56000,

  EBaud57600 = CBR_57600, **EBaud115200** = CBR_115200, **EBaud128000** = CBR_128000, **EBaud256000** = CBR_256000,

  EBaud500000 = 500000, **EBaudCSM9600** = 2150, **EBaudCSM19200** = 4301, **EBaudCSM38400** = 8602,

  EBaudCSM500000 = 115000 }
- enum **EDataBits** {

  EDataUnknown = -1, **EData5** = 5, **EData6** = 6, **EData7** = 7,

  EData8 = 8 }
- enum **EParity** {

  EParUnknown = -1, **EParNone** = NOPARITY, **EParOdd** = ODDPARITY, **EParEven** = EVENPARITY,

  EParMark = MARKPARITY, **EParSpace** = SPACEPARITY }
- enum **EStopBits** { **EStopUnknown** = -1, **EStop1** = ONESTOPBIT, **EStop1_5** = ONE5STOPBITS, **EStop2** = TWOSTOPBITS }
- enum **EHandshake** { **EHandshakeUnknown** = -1, **EHandshakeOff** = 0, **EHandshakeHardware** = 1, **EHandshakeSoftware** = 2 }

- enum **EReadTimeout** { **EReadTimeoutUnknown** = -1, **EReadTimeoutNonblocking** = 0, **EReadTimeoutBlocking** = 1 }
- enum **EError** {

  EErrorUnknown = 0, **EErrorBreak** = CE_BREAK, **EErrorFrame** = CE_FRAME, **EErrorIOE** = CE_IOE,

  EErrorMode = CE_MODE, **EErrorOverrun** = CE_OVERRUN, **EErrorRxOver** = CE_RXOVER, **EErrorParity** = CE_RXPARITY,

  EErrorTxFull = CE_TXFULL }
- enum **EPort** { **EPortUnknownError** = -1, **EPortAvailable** = 0, **EPortNotAvailable** = 1, **EPortInUse** = 2 }

Public Member Functions

- EPort **CheckPort** (LPCTSTR lpszDevice)
- virtual LONG **Open** (LPCTSTR lpszDevice, DWORD dwInQueue=2048, DWORD dwOutQueue=2048)
- virtual LONG **ClosePort** (void)
- virtual LONG **Setup** (EBaudrate eBaudrate=EBaud9600, EDataBits eDataBits=EData8, EParity eParity=EParNone, EStopBits eStopBits=EStop1)
- virtual LONG **SetEventChar** (BYTE bEventChar, bool fAdjustMask=true)
- virtual LONG **SetMask** (DWORD dwMask=EEventBreak|EEventError|EEventRecv)
- virtual LONG **WaitEvent** (LPOVERLAPPED lpOverlapped=0, DWORD dwTimeout=INFINITE)
- virtual LONG **SetupHandshaking** (EHandshake eHandshake)
- virtual LONG **SetupReadTimeouts** (EReadTimeout eReadTimeout)
- virtual EBaudrate **GetBaudrate** (void)
- virtual EDataBits **GetDataBits** (void)
- virtual EParity **GetParity** (void)
- virtual EStopBits **GetStopBits** (void)
- virtual EHandshake **GetHandshaking** (void)
- virtual DWORD **GetEventMask** (void)
- virtual BYTE **GetEventChar** (void)
- virtual LONG **Write** (const void *pData, size_t iLen, DWORD *pdwWritten=0, LPOVERLAPPED lpOverlapped=0, DWORD dwTimeout=INFINITE)
- virtual LONG **Write** (LPCSTR pString, DWORD *pdwWritten=0, LPOVERLAPPED lpOverlapped=0, DWORD dwTimeout=INFINITE)
- virtual LONG **NTRead** (void *pData, size_t iLen, DWORD *pdwRead=0, LPOVERLAPPED lpOverlapped=0, DWORD dwTimeout=INFINITE)
- EEvent **GetEventType** (void)
- EError **GetError** (void)
- HANDLE **GetCommHandle** (void)
- bool **IsOpen** (void) const
- LONG **GetLastError** (void) const
- bool **GetCTS** (void)
- bool **GetDSR** (void)
- bool **GetRing** (void)
- bool **GetRLSD** (void)
- virtual int **Flush** (void)

Protected Attributes

- LONG **m_lLastError**
- HANDLE **m_hFile**
- EEvent **m_eEvent**
- HANDLE **m_hevtOverlapped**

Classes

- class **CDCB**

4.6.1 Detailed Description

Middle Layer class for Windows Serial port (c) Ramon de Klein.

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

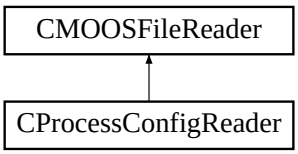
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/NTSerial.h
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/NTSerial.cpp

4.7 CProcessConfigReader Class Reference

Class for reading MOOS configuration files.

`#include <ProcessConfigReader.h>`

Inheritance diagram for CProcessConfigReader::



Public Member Functions

- std::string **GetAppName** ()
- std::string **GetFileName** ()
- void **SetAppName** (std::string sAppName)
- bool **GetConfigurationParam** (std::string sAppName, std::string sParam, std::string &sVal)

  READ STRINGS.

- bool **GetConfigurationParam** (std::string sAppName, std::string sParam, double &dfVal)
- bool **GetConfigurationParam** (std::string sAppName, std::string sParam, bool &bVal)
- bool **GetConfigurationParam** (std::string sAppName, std::string sParam, int &nVal)
- bool **GetConfigurationParam** (std::string sAppName, std::string sParam, std::vector< double > &Vec, int &nRows, int &nCols)
- bool **GetConfigurationParam** (std::string sParam, std::string &sVal)
- bool **GetConfigurationParam** (std::string sParam, double &dfVal)

  READ DOUBLES.

- bool **GetConfigurationParam** (std::string sParam, bool &bVal)

  READ BOOLS.

- bool **GetConfigurationParam** (std::string sParam, int &nVal)

  READ INTS.

- bool **GetConfigurationParam** (std::string sParam, std::vector< double > &Vec, int &nRows, int &nCols)

  READ VECTORS.

- bool **GetConfiguration** (std::string sAppName, STRING_LIST &Params)

Public Attributes

- std::string **m_sAppName**

4.7.1 Detailed Description

Class for reading MOOS configuration files.

4.7.2 Member Function Documentation

4.7.2.1 std::string CProcessConfigReader::GetAppName ()

returns the name of the application an instance of this class is concerned with

4.7.2.2 bool CProcessConfigReader::GetConfiguration (std::string *sAppName*, STRING_LIST & *Params*)

return a list of strings of Token = Val for the specfied named application configuration block

4.7.2.3 bool CProcessConfigReader::GetConfigurationParam (std::string *sParam*, std::vector< double > & *Vec*, int & *nRows*, int & *nCols*)

READ VECTORS.

read a vector<double> parameter for a Process "m_sName" (can be interprested as a matrix with (rows x cols)

4.7.2.4 bool CProcessConfigReader::GetConfigurationParam (std::string *sParam*, int & *nVal*)

READ INTS.

read a int parameter for a Process "m_sName"

4.7.2.5 bool CProcessConfigReader::GetConfigurationParam (std::string *sParam*, bool & *bVal*)

READ BOOLS.

read a bool parameter for a Process "m_sName"

4.7.2.6 bool CProcessConfigReader::GetConfigurationParam (std::string *sParam*, double & *dfVal*)

READ DOUBLES.

read a double parameter for a Process "m_sName"

4.7.2.7 bool CProcessConfigReader::GetConfigurationParam (std::string *sParam*, std::string & *sVal*)

read a string parameter for a Process "m_sName"

4.7.2.8 bool CProcessConfigReader::GetConfigurationParam (std::string *sAppName*, std::string *sParam*, std::vector< double > & *Vec*, int & *nRows*, int & *nCols*)

read a vector<double> parameter for a named process

4.7.2.9 bool CProcessConfigReader::GetConfigurationParam (std::string *sAppName*, std::string *sParam*, int & *nVal*)

read a integer parameter for a named process

4.7.2.10 bool CProcessConfigReader::GetConfigurationParam (std::string *sAppName*, std::string *sParam*, bool & *bVal*)

read a bool parameter for a named process

4.7.2.11 bool CProcessConfigReader::GetConfigurationParam (std::string *sAppName*, std::string *sParam*, double & *dfVal*)

read a string parameter for a named process

4.7.2.12 bool CProcessConfigReader::GetConfigurationParam (std::string *sAppName*, std::string *sParam*, std::string & *sVal*)

READ STRINGS.

read a string parameter for a named process

4.7.2.13 std::string CProcessConfigReader::GetFileName ()

returns the name of the mission file this process is accessing

4.7.2.14 void CProcessConfigReader::SetAppName (std::string *sAppName*)

set the name of the application (MOOSProcess) that this class shoud concern itself with (unless directed otherwise)

4.7.3 Member Data Documentation

4.7.3.1 std::string CProcessConfigReader::m_sAppName

the name of process an instance this class will handle unless told otherwise

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

- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/**ProcessConfigReader.h**
- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/ProcessConfigReader.cpp

4.8 dynamic_caster< D > Struct Template Reference

`#include <MOOSGenLibGlobalHelper.h>`

Public Member Functions

- template<class S> D **operator()** (S s) const

4.8.1 Detailed Description

template<class D> struct dynamic_caster< D >

Functor class for performing dynamic_cast between two types. Use it with stl::transform when copying between two collections with different element types

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

- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/**MOOSGenLibGlobalHelper.h**

4.9 static_caster< D > Struct Template Reference

`#include <MOOSGenLibGlobalHelper.h>`

Public Member Functions

- template<class S> D **operator()** (S s) const

4.9.1 Detailed Description

template<class D> struct static_caster< D >

Functor class for performing static_cast between two types. Use it with stl::transform when copying between two collections with different element types

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

- /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/**MOOSGenLibGlobalHelper.h**

Chapter 5

MOOSGenLib File Documentation

5.1 /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSGenLibGlobalHelper.h File Reference

```
#include <string>
#include <list>
#include <vector>
#include <algorithm>
```

Classes

- struct **static_caster< D >**
- struct **dynamic_caster< D >**

Defines

- #define **PI** 3.141592653589
- #define **MOOSHERE** MOOSFormat("File %s Line %d", __FILE__,__LINE__).c_str()
- #define **UNUSED_PARAMETER**(a)

Typedefs

- typedef std::list< std::string > **STRING_LIST**

Functions

- bool **MOOSGetValueFromToken** (STRING_LIST &sParams, const std::string &sToken, std::string &sVal)
- bool **MOOSValFromString** (std::string &sVal, const std::string &sStr, const std::string &sTk)
- bool **MOOSValFromString** (double &dfVal, const std::string &sStr, const std::string &sTk)

- bool **MOOSValFromString** (int &nVal, const std::string &sStr, const std::string &sTk)
- bool **MOOSValFromString** (bool &bVal, const std::string &sStr, const std::string &sTk)
- bool **MOOSValFromString** (std::vector< double > &dfValVec, int &nRows, int &nCols, const std::string &sStr, const std::string &sToken)
- bool **MOOSVectorFromString** (const std::string &sStr, std::vector< double > &dfVecVal, int &nRows, int &nCols)
- bool **MOOSVectorFromString** (const std::string &sStr, std::vector< unsigned int > &dfVecVal, int &nRows, int &nCols)
- std::string **DoubleVector2String** (const std::vector< double > &V)
- std::stringstream & **Write** (std::stringstream &os, const std::vector< double > &Vec)
- std::stringstream & **Write** (std::stringstream &os, const std::vector< int > &Vec)
- std::string **MOOSChomp** (std::string &sStr, const std::string &sTk=",")
- void **MOOSRemoveChars** (std::string &sStr, const std::string &sTok)
- void **MOOSToUpper** (std::string &str)
- void **MOOSTrimWhiteSpace** (std::string &str)
- bool **MOOSIsNumeric** (std::string str)
- bool **MOOSStrCmp** (std::string s1, std::string s2)
- double **GetMOOSSkew** ()
- void **SetMOOSSkew** (double dfSkew)
- void **MOOSPause** (int nMS)
- double **MOOSTime** ()
- double **HPMOOSTime** ()
- int **MOOSGetch** ()
- void **MOOSTrace** (std::string Str)
- void **MOOSTrace** (char *FmtStr,...)
- std::string **MOOSFormat** (char *FmtStr,...)
- bool **MOOSFail** (char *FmtStr,...)
- std::string **MOOSGetTimeStampString** ()
- std::string **MOOSGetDate** ()
- void **Progress** (double dfPC)
- std::string **MOOSThirdPartyActuationString** (double *pdfRudder, double *pdfElevator, double *pdfThrust)
- std::string **MOOSThirdPartyStatusString** (std::string sStatusCommand)
- double **MOOS_ANGLE_WRAP** (double dfAng)
- double **MOOSDeg2Rad** (double dfDeg)
- double **MOOSRad2Deg** (double dfRad)
- bool **MOOSAbsLimit** (double &dfVal, double dfLimit)
- double **MOOSWhiteNoise** (double Sigma)
- double **MOOSNormalInv** (double dfArea)
- int **MOOSDiscreteUniform** (int nMin, int nMax)
- double **MOOSUniformRandom** (double dfMin, double dfMax)
- template<class T> const T & **MOOSClamp** (const T &val, const T &min, const T &max)
- bool **GetDirectoryContents** (const std::string &sPath, std::list< std::string > &sContents, bool bFiles=true)
- bool **MOOSCreateDirectory** (const std::string &sDirectory)
- bool **MOOSFileParts** (std::string sFullPath, std::string &sPath, std::string &sFile, std::string &sExtension)
- template<class T> T **SwapByteOrder** (const T &v)
- bool **IsLittleEndian** ()

5.1.1 Detailed Description

5.1.2 Define Documentation

5.1.2.1 #define MOOSHERE MOOSFormat("File %s Line %d", __FILE__,__LINE__).c_str()

useful macro for debugging prints line and file

5.1.3 Function Documentation

5.1.3.1 std::string DoubleVector2String (const std::vector< double > & *V*)

formats a vector of doublse into standard MOOS format

5.1.3.2 bool GetDirectoryContents (const std::string & *sPath*, std::list< std::string > & *sContents*, bool *bFiles*)

returns a string list of directories or files in a specified location exludes . and ..

5.1.3.3 double GetMOOSSkew ()

return the offset between DB time and client time

5.1.3.4 double HPMOOSTime ()

return high precision timestamp - time since unix in seconds

5.1.3.5 bool IsLittleEndian ()

returns true if architecture is LittleEndian (true for x86 Architectures) Note after first call it remembers answer in a static so v. littel overhead in calling this function frequently

5.1.3.6 double MOOS_ANGLE_WRAP (double *dfAng*)

Bound angle to +/-PI

5.1.3.7 bool MOOSAbsLimit (double & *dfVal*, double *dfLimit*)

limit dfVal to lie between +/- dfLimit)

5.1.3.8 template<class T> const T& MOOSClamp (const T & *val*, const T & *min*, const T & *max*)

Clamps a templated type between two values

5.1.3.9 bool MOOSCreateDirectory (const std::string & *sDirectory*)

make a directory

5.1.3.10 double MOOSDeg2Rad (double *dfDeg*)

convert deg to rad

5.1.3.11 int MOOSDiscreteUniform (int *nMin*, int *nMax*)

generates uniform noise in integers between interval nMin->nMax

5.1.3.12 bool MOOSFail (char * *FmtStr*, ...)

like MOOSTrace but returns false - useful for return statements

5.1.3.13 bool MOOSFileParts (std::string *sFullPath*, std::string & *sPath*, std::string & *sFile*, std::string & *sExtension*)

splits a fully qualified path into parts -path, filestem and extension

5.1.3.14 std::string MOOSFormat (char * *FmtStr*, ...)

return a formatted string (with printf-like format codes

5.1.3.15 int MOOSGetch ()

useful keyboard trap

5.1.3.16 std::string MOOSGetDate ()

get teh current date formatted nicely

5.1.3.17 std::string MOOSGetTimeStampString ()

return nicely formatted time stamp string

5.1.3.18 bool MOOSIsNumeric (std::string *str*)

returbn true if numeric

5.1.3.19 double MOOSNormalInv (double *dfArea*)

returns x for probablity mass such p(v<=x) = dfArea)

5.1.3.20 void MOOSPause (int *nMS*)

pause for nMS milliseconds

5.1.3.21 double MOOSRad2Deg (double *dfRad*)

convert rad 2 deg

5.1.3.22 void MOOSRemoveChars (std::string & *sStr*, const std::string & *sTok*)

remove all characters in sTok from sStr

5.1.3.23 bool MOOSStrCmp (std::string *s1*, std::string *s2*)

case insensitive string comparison. returns true if equal

5.1.3.24 double MOOSTime ()

return time as a double (time since unix in seconds)

5.1.3.25 void MOOSToUpper (std::string & *str*)

convert string to upper case

5.1.3.26 void MOOSTrace (char * *FmtStr*, ...)

print a formatted string (with printf-like format codes) and to debug window in DevStudio

5.1.3.27 void MOOSTrace (std::string *Str*)

print a string

5.1.3.28 void MOOSTrimWhiteSpace (std::string & *str*)

remove white space form start and end of a string

5.1.3.29 double MOOSUniformRandom (double *dfMin*, double *dfMax*)

generates uniform noise in interval dfMin-dfMax

5.1.3.30 double MOOSWhiteNoise (double *Sigma*)

returns sample fom Gaussian process strength Sigma mena zero

5.1.3.31 void Progress (double *dfPC*)

prints a "progress bar" upto 40 characters long dfPC is the fraction complete - ie 0:1

5.1.3.32 template<class T> T SwapByteOrder (const T & *v*)

templated function which swaps byte order of type T returning it

5.1.3.33 std::stringstream& Write (std::stringstream & *os*, const std::vector< int > & *Vec*)

formats a vector of ints into standard MOOS format

5.1.3.34 std::stringstream& Write (std::stringstream & *os*, const std::vector< double > & *Vec*)

formats a vector of doubles into standard MOOS format

5.2 /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/MOOSSerialPort.h File Reference

```
#include "MOOSLock.h"
#include <string>
#include <list>
```

Classes

- class **CMOOSSerialPort**

  Cross Platform Serial Port Base Class.

- class **CMOOSSerialPort::CMOOSSerialTelegram**

Defines

- #define **MOOSSERIALPORTH**
- #define **DEFAULT_PORT** "/dev/ttyS0"
- #define **DEFAULT_BAUDRATE** 9600
- #define **TELEGRAM_LEN** 1000

Typedefs

- typedef std::list< std::string > **STRING_LIST**

5.2.1 Detailed Description

5.3 /home/pnewman/code/MOOS/trunk/Core/MOOSGenLib/ProcessConfigReader.h File Reference

```
#include "MOOSFileReader.h"
#include <string>
#include <list>
#include <vector>
```

Classes

- class **CProcessConfigReader**
  Class for reading MOOS configuration files.

Typedefs

- typedef std::list< std::string > **STRING_LIST**

5.3.1 Detailed Description

Index