MiscXmodem

Inherits From: Object

Declared In: **<**misckit/MiscXmodem.h>

Class Description

The MiscXmodem class implements the XMODEM, XMODEM-CRC and XMODEM-1k-CRC serial-line file transfer prtocols.   A complete description of these protocols can be found in, *XMODEM/YMODEM PROTOCOL REFERENCE -- A compendium of documents describing the XMODEM and YMODEM File Transfer Protocols*, Chuck Forsberg, Omen Technology Incorporated, Portland, Oregon, 1988.

Objects of this class do not modify the serial line settings in any way.   Before invoking the data transfer methods (**sendSerialPort:toStream:**, **sendStream:toSerialPort:**) the serial line must be configured to provide an 8-bit-clean path at the appropriate speed.

MiscXmodem objects are multithreaded but the sub-thread makes no Objective-C calls, so applications using these objects do not have to call **objc_setMultithreaded**().   Applications should however use **cthread_errno**() rather than **errno**, at least while data transfer is in progress.

Instance Variables

BOOL **isActive;**
BOOL **isText;**
BOOL **use1kPackets;**
BOOL **isDynamicCheck;**
BOOL **useCRC;**
MiscSerialPort ***serialPort;**
BOOL **serialPortWasSuspended;**
int **fd;**
fd_set **checkFdSet;**
NXStream ***stream;**
cthread_t **xmodemThread;**
port_t **messagePort;**
port_t **replyPort;**
id **delegate;**
unsigned char **packetBuffer[1024+5];**
unsigned char **smallPacketData[1024];**
int **smallPacketIndex;**

```
int smallPacketCount;
BOOLnewlineFlag;
BOOL isDynamicCheckPacket;
int lastPacketNumber;
int goodPacketCount;
float kbytes;
```

isActive	YES if a data transfer operation in in progress.
isText	YES if data will be treated as ASCII text.
use1kPackets	YES if data transmission will use large packets.
isDynamicCheck	YES if error checking method will be determined dynamically.
useCRC	YES if 16-bit CRC error checking method is in use.
MiscSerialPort	Serial port object to be used for data transfer.
serialPortWasSuspended	Remember whether or not to **continue** serial port object when data transfer is complete.
fd	Serial line file descriptor.
checkFdSet	Select(2) file descriptor bitmap.
stream	The stream to be used for data transfer.

xmodemThread	Data transfer thread.
messagePort	Mach port for messages from data transfer thread to main thread.
replyPort	Mach port for messages from main thread to data transfer thread.
delegate	The object that responds to notification messages.
packetBuffer	Current incoming or outgoing packet.
smallPacketData	Storage for small packet transmission.
smallPacketIndex	Index of next small packet data item to transmit.
smallPacketCount	Number of data bytes in smallPacketData.
newlineFlag	YES if a newline character needs to be transmitted (text mode data transfers).
isDynamicCheckPacket	YES if the next packet will be used to determine the error-checking method.
lastPacketNumber	Packet number (0-255) of most recent successfully-received packet.
goodPacketCount	Number of packets successfully transferred.
kbytes	Number of kbytes of data successfully transferred.

Method Types

Initializing and freeing instances - init

- free

Data transfer options
- setFileTypeText:
- xmodemSet1kPackets:
- xmodemSetCheckTypeDynamic:
- xmodemSetCheckTypeCRC:
- isFileTypeText
- xmodemIs1kPackets
- xmodemIsCheckTypeDynamic
- xmodemIsCheckTypeCRC

Data transfer operations
- sendToSerialPort:fromStream:
- sendToStream:fromSerialPort:
- isActive

Delegate
- setDelegate:
- delegate
- didTransferPacket:kbytes:
- didRetryNumber:
- didFinish:

Instance Methods

delegate
    -**delegate**

Returns the object's delegate.

free
    -**free**

Cancels the data transfer operation, if any, and frees the object.

init
    -**init**

Initializes a new MiscXmodem instance.   The default settings are text files, 1k packets, and dynamic selection of error checking method.

setDelegate:
    -**setDelegate:***id*

Sets the object's delegate.   Delegate objects are sent messages describing the progress of data transfer operations.

See also: - didFinish:, - didTransferPacket:kbytes:,   - didRetryNumber:

didFinish:
   -**didFinish:**(const char *)*failureCode*

Sent to the delegate when data transfer operation completes.   If the transfer completed successfully, the *failureCode* string will be empty (""), otherwise the string will describe the condition that caused the transfer to fail.

didRetryNumber:
   -**didRetryNumber:**(int)*anInt*

Sent to the delegate when the data transfer operation retries a packet transfer.   The argument is the number of times that the current packet has been retried.

didTransferPacket:kbytes:
   -**didTransferPacket:**(int)*anInt* **kbytes:**(float)*aFloat*

Sent to the delegate when the a packet is successfully transferred.   The arguments are the number of packets and the amount of data (in kbytes) that have been transferred so far.

xmodemIs1kPackets
   -(BOOL)**xmodemIs1kPackets**

Returns YES if data transmission will use 1k packets, NO if it will use only 128-byte packets. The default value is YES.

See also: - xmodemSet1kPackets:

isActive
  -(BOOL)**isActive**

Returns YES if the MiscXmodem object is in the process of transferring data between the seria port and the stream, NO if it is not.

See also: - sendFromSerialPort:toStream:, - sendFromStream:toSerialPort:, - xmodemIsDone:

xmodemIsCheckTypeCRC
  -(BOOL)**xmodemIsCheckTypeCRC**

Returns YES if data reception will use CRC error checking, NO if it will use a checksum. The default value is YES.

See also: - xmodemSet1kPackets:, - xmodemSetDynamicCheck:

xmodemIsCheckTypeDynamic

-(BOOL)**xmodemIsCheckTypeDynamic**

Returns YES if data reception will first attempt to use CRC error checking and automatically revert to use a checksum if the sender does not support CRC error checking.   Returns NO if data reception will not dynamically adapt to the sender's preference.   The default value is YES.

See also: - xmodemSetDynamicCheck, - xmodemSet1kPackets

isFileTypeText
-(BOOL)**isFileTypeText**

Returns YES if data transfer will convert files to/from DOS text format.   Returns NO if data will be transferred unmodified.

See also: - setFileTypeText

sendToStream:fromSerialPort:
-**sendToStream:**(NXStream *)*aStream* **fromSerialPort:**(int)*ttyFd*

Spawns a subthread to perform a data reception operation and returns **self** if a transfer is not already in progress. Does nothing, and returns **nil**, if a transfer is already in progress.   The application must not use the stream or the tty until the transfer is complete.

See also: - isActive, - didFinish:, - didTransferPacket:kbytes:,   - didRetryNumber:

sendToSerialPort:fromStream:
 -**sendToSerialPort:**(int)*ttyFd* **fromStream:**(NXStream *)*aStream*

Spawns a subthread to perform a data transmission operation and returns **self** if a transfer is not already in progress. Does nothing, and returns **nil**, if a transfer is already in progress.   The application must not use the stream or the tty until the transfer is complete.

See also: - isActive, - didFinish:, - didTransferPacket:kbytes:,   - didRetryNumber:

xmodemSet1kPackets:
 -**xmodemSet1kPackets:**(BOOL)*flag*

Control the use of 1k packets when transmitting.   If *flag* is YES, 1-kbyte packets will be transmitted until there are fewer than 1024 bytes left to send at which point 128-byte packets will be used.   If *flag* is NO, only 128-byte packets will be used. Both 128-byte and 1-kbyte packets are always accepted when receiving.

See also: - xmodemIs1kPackets

xmodemSetCheckTypeCRC:
 -**xmodemSetCheckTypeCRC:**(BOOL)*flag*

Select the error checking method to be used.   If *flag* is YES, a 16-bit CRC error checking code will be used.   If

flag is NO, a simple 8-bit checksum will be used.   This option is used only when receiving a file, since data transmission operations always adapts to the check method chosen by the receiver.

See also: - xmodemIsCheckTypeCRC, -xmodemSetCheckTypeDynamic

xmodemSetCheckTypeDynamic:
  -**xmodemSetCheckTypeDynamic:**(BOOL)*flag*

If *flag* is YES, data reception operations will first try to use a 16-bit CRC error checking code.   If the sender does not support CRC, the receiver will automatically revert to an 8-bit checksum.   If *flag* is YES, data reception operations will use the check method specified by **xmodemSetCheckTypeCRC:**.

See also: - xmodemIsCheckTypeDynamic, -xmodemSetCheckTypeCRC:

setFileTypeText:
  -**setFileTypeText:**(BOOL)*flag*

Controls the interpretation of transferred data.   If *flag* is NO, data will be transferred unmodified.   Data will be padded to a 128-byte boundary so the contents of a data transfer operation can grow by up to 127 bytes.   If *flag* is YES, data will be transferred in DOS text format.   On transmission, <newline> characters will be converted to <carriage return><newline> pairs and a control-Z character (ASCII 032) will be used to mark the end of the data. On reception,
<carriage return> characters will be removed and a control-Z and subsequent characters will not be added to the stream.

See also: - isFileTypeText