

# MiscXmodem

**Inherits From:** MiscSerialPort

**Declared In:** MiscKit/MiscXmodem.h

## Class Description

The MiscXmodem class implements the XMODEM, XMODEM-CRC and XMODEM-1k-CRC serial-line file transfer protocols. 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 (**xmodemStartSending:**, **xmodemStartReceiving**) the serial line must be configured to provide an 8-bit-clean path at the appropriate speed.

MiscXmodem objects override their superclass's **\_serialDataArrived:** method to intercept incoming characters during a data transfer operation. When a data transfer operation is not in progress MiscXmodem objects behave like MiscSerialPort objects.

Each packet successfully transferred is reported by the posting of a **MiscXmodemPacketNotification**. Each retry operation is reported by the posting of a **MiscXmodemRetryNotification**. On completion of the data transfer operation, a **MiscXmodemCompletionNotification** is posted. All notifications are posted to the default message center. In all cases, the *object* associated with the notification is the MiscXmodem object which posted the notification. The *userinfo* dictionary depends upon the particular notification.

## Instance Variables

```
BOOL isText;  
BOOL use1kPackets;  
BOOL useDynamicCheck;  
BOOL useCRC;  
BOOL isReceiving;  
NSMutableData *data;  
unsigned char pkBuffer[1024+5];  
int lastPacketNumber;  
int pkIndex;  
int pkCount;
```

```
BOOL transferInProgress;  
int transferPacketCount;  
int transferByteCount;  
int transferRetryCount;  
int nakCount;  
BOOL rxExpectPacket;  
BOOL rxSendDynamicCheck;  
BOOL rxCRC;  
BOOL tx1kPackets;  
BOOL txNewline;  
BOOL txCRC;  
unsigned int txCount;  
NSTimer *timer;
```

isText	Controls whether data will be transferred as ASCII text or as a binary image.
use1kPackets	Controls size of transmitted packets. If <b>YES</b> , 1 kbyte packets be used to transfer data, otherwise 128 byte packets will be used.
useDynamicCheck	If <b>YES</b> , the object will begin a data reception operation by adapting to the packet check method supported by the transmitter. If <b>NO</b> , the packet check method is chosen by <i>useCRC</i> .
useCRC	If the packet check method is statically selected this variable controls whether the check method will be a 2-byte CRC or a single-byte checksum.
isReceiving	Direction of data transfer.

data	Data to be sent or data that has been received.
pkBuffer[1024+5]	Storage for incoming or outgoing packet.
lastPacketNumber	The number of the last packet received or transmitted.
pkIndex	Index into <i>pkBuffer</i> .
pkCount	Index into <i>pkBuffer</i> .
transferInProgress	<b>YES</b> when a data transfer operation is in progress.
transferPacketCount	Number of packets transferred.
transferByteCount	Number of bytes transferred.
transferRetryCount	Number of retry operations.
nakCount	Number of NAK's sent or received.
rxExpectPacket	Receiver is expecting a packet.
rxSendDynamicCheck	Receiver is requesting that transmitter use CRC packet check method.
rxCRC	Packet check method used during data reception operation.
tx1kPackets	Control size of transmitted packets.
txNewline	Used during expansion of <newline> into <carriage return><newline> during ASCII text transmission.
txCRC	Packet check method used during transmission operation.

txCount	Number of bytes taken from <i>data</i> .
timer	Packet timer.

## Method Types

Starting a transfer	<ul style="list-style-type: none"><li>- xmodemStartSending:</li><li>- xmodemStartReceiving</li></ul>
Stopping a transfer in progress	<ul style="list-style-type: none"><li>- xmodemCancelTransfer</li></ul>
Transfer attributes	<ul style="list-style-type: none"><li>- xmodemIs1kPackets</li><li>- xmodemIsCheckTypeCRC</li><li>- xmodemIsCheckTypeDynamic</li><li>- xmodemIsFileTypeText</li><li>- xmodemSet1kPackets:</li><li>- xmodemSetCheckTypeCRC:</li><li>- xmodemSetCheckTypeDynamic:</li><li>- xmodemSetFileTypeText:</li></ul>
Transfer statistics	<ul style="list-style-type: none"><li>- xmodemTransferByteCount</li><li>- xmodemTransferPacketCount</li><li>- xmodemTransferRetryCount</li></ul>

## Instance Methods

### **xmodemCancelTransfer**

- (void)xmodemCancelTransfer

Cancel a data transfer operation. The sender or receiver at the other end of the line is not informed that the transfer has been cancelled, but should time out after a while.

**See also:** - xmodemStartSending:, - xmodemStartReceiving

### **xmodemIs1kPackets**

- (BOOL)xmodemIs1kPackets

Returns **YES** if data transmission will use 1 kbyte packets. Returns **NO** if data transmission will use 128 byte packets.

**See also:** - xmodemSet1kPackets:

### **xmodemIsCheckTypeCRC**

- (BOOL)xmodemIsCheckTypeCRC

Returns **YES** if data reception will use CRC packet verification. Returns **NO** if data reception will use checksum packet verification.

**See also:** - `xmodemSetCheckTypeCRC:`, - `xmodemSetCheckTypeDynamic:`

### **`xmodemIsCheckTypeDynamic`**

- (BOOL)`xmodemIsCheckTypeDynamic`

Returns **YES** if object will attempt to negotiate packet check method with the transmitter at the beginning of a data reception operation. Returns **NO** if the object will receive packets using the check method chosen by the `xmodemSetCheckTypeCRC:` method.

**See also:** - `xmodemSetCheckTypeDynamic:`

### **`xmodemIsFileTypeText`**

- (BOOL)`xmodemIsFileTypeText`

Returns **YES** if data will be transferred as ASCII text files. Returns **NO** if data will be transferred as a binary image. On transmission, <newline> characters will be converted to <carriage return><newline> pairs. On reception, <carriage return><newline> pairs will be converted to <newline> characters. A control-Z character marks the end of the data.

**See also:** - `xmodemSetFileTypeText:`

### **`xmodemSet1kPackets:`**

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

Controls the size of transmitted packets. If *flag* is **YES** 1 kbyte packets will be transmitted. If *flag* is **NO** 128 byte packets will be transmitted. The default value is **YES**.

**See also:** - **xmodemIs1kPackets**

#### **xmodemSetCheckTypeCRC:**

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

Control the method used to verify incoming packets. If *flag* is **YES** packets will be verified using a 2-byte CRC. If *flag* is **NO** packets will be verified using a 1-byte checksum. Note that this selection is ignored if dynamic packet check selection is enabled. The default value is **YES**.

**See also:** - **xmodemIsCheckTypeCRC**

#### **xmodemSetCheckTypeDynamic:**

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

Control the negotiation of packet verification at the beginning of a data reception operation. If *flag* is **YES** the object will attempt to negotiate the packet verification method with the transmitter at the beginning of a data reception operation. If *flag* is **NO** incoming packets will be verified using the method chosen by the **xmodemSetCheckTypeCRC**: method. The default value is **YES**.

The packet verification method used during data transmission operations is always dynamically selected and is controlled by the receiver at the other end of the wire.

**See also:** - **xmodemIsCheckTypeDynamic**

### **xmodemSetFileTypeText:**

- (void)xmodemSetFileTypeText:(BOOL)*flag*

Controls the format of data transmission and reception. If *flag* is **YES**, on transmission, <newline> characters will be converted to <carriage return><newline> pairs. On reception, <carriage return><newline> pairs will be converted to <newline> characters. A control-Z character in a packet marks the end of the data. If *flag* is **NO** data will be transferred as a binary image. The data transferred will be padded to an even multiple of 128 bytes.

The default value is for binary transfers.

**See also:** - **xmodemIsFileTypeText**

### **xmodemStartReceiving**

- (BOOL)xmodemStartReceiving

Start a data reception operation.

**See also:** - **xmodemCancelTransfer**

**xmodemStartSending:**

- (BOOL)xmodemStartSending:(NSData \*)dataSource

Start sending the contents of *dataSource*.

**See also:** - xmodemCancelTransfer

**xmodemTransferByteCount**

- (unsigned int)xmodemTransferByteCount

Return the number of bytes transferred in the current, or most recent, data transfer.

**See also:** - xmodemTransferPacketCount

**xmodemTransferPacketCount**

- (unsigned int)xmodemTransferPacketCount

Return the number of packets transferred in the current, or most recent, data transfer.

**See also:** - xmodemTransferByteCount

**xmodemTransferRetryCount**

- (unsigned int)xmodemTransferRetryCount

Return the number of retry operations in the current, or most recent, data transfer.

## Notifications

### **MiscXmodemCompletionNotification**

Posted when a data transfer operation completes or is cancelled.

The notification contains:

<b>Notification Object</b>	The notifying MiscXmodem object.
<b>Userinfo Key</b>	<b>Value</b>
MiscXmodemCompletionNotificationDataItem	An <b>NSData</b> object containing the received data. Present only if the operation was a data reception and if the data reception succeeded.
MiscXmodemCompletionNotificationFailureReason	An <b>NSString</b> object describing the condition that caused the unsuccessful termination of a data transfer operation. Present only when a data transfer fails.

### **MiscXmodemPacketNotification**

Posted when a packet is successfully transferred.

The notification contains:

<b>Notification Object</b>	The notifying MiscXmodem object.
----------------------------	----------------------------------

**Userinfo**

None

**MiscXmodemRetryNotification**

Posted when a packet transfer must be retried.

The notification contains:

**Notification Object**

The notifying MiscXmodem object.

**Userinfo**

None