

XtraNet v1.02

Documentation

copyright © 1996-97 by g/matter, inc.

Written by Terry R. Schussler

Table of Contents

Table of Contents

Overview of XtraNet

The Four Types of Xtras

What is a Sprite Xtra?

What is a Lingo Xtra?

What is a Transition Xtra?

What is a Tool Xtra?

Version History

1.0

1.01

1.02

XtraNet Installation and End-user Delivery

How To Install XtraNet with Director

How To Install XtraNet with Shockwave

Shipping XtraNet with a Runtime Projector

Delivering XtraNet via Shockwave

Getting Started with XtraNet

Using the Xtras Menu

Using Lingo

Registering XtraNet

Using an Unregistered Copy of XtraNet

The XtraNet Lingo Interface

Overview

Standard Lingo Xtra Functions/Commands

interface(xtra "XtraNet") -or- mMessageList(xtra "XtraNet")

General

XNetInit

XNetIdle

XNetClose integer cookieID

XNetShutdown

XNetSetProxy string hostName, integer portAddress

Informational

XNetVersion()

XNetIPAddress()

XNetPort(integer cookieID)

XNetDNSLookup(string hostname|string IPAddress)

XNetError(integer errorCode)
 XNetSetConnectionLoc point upperLeftCornerofDialog
 XNetSetConnectionMsg string dialogText

Operation Timing

XNetSetDefaultTimeout integer timeOut
 XNetGetDefaultTimeout
 XNetSetTimeout integer cookie, integer timeOut
 XNetGetTimeout integer cookie

Netscape-style Cookies

XNetSetCookie string cookie
 XNetClearCookie

HTTP/FTP Access

XNetGetFile string url, string filename, string callback
 XNetGetGlobal string url, string globalname, string callback
 XNetPutFile string filename, string url, string callback
 XNetPutGlobal string globalname, string url, string callback

Peer Class

XNetPeerCreate
 XNetPeerConnect integer cookieID, string remoteHost, integer port
 XNetPeerAccept integer cookieID, integer port
 XNetPeerGet integer cookieID
 XNetPeerSend integer cookieID, string theText

Client Class

XNetClientCreate
 XNetClientConnect integer cookieID, string remoteHost, integer port
 XNetClientAccept integer cookieID, integer port
 XNetClientGet integer cookieID
 XNetClientSend integer cookieID, string theText

Server Class

XNetServerStart integer maxClients, integer port, string callback
 XNetServerSend integer cookieID, string theText [, string clientName]
 XNetServerDisconnect integer cookieID, string clientName

Raw Class

XNetRawCreate
 XNetRawConnect integer cookie, string remoteHost, integer port
 XNetRawAccept integer cookie, integer port
 XNetRawGet integer cookie
 XNetRawSend integer cookie, string theText

XtraNet's Callback Mechanism

The Callback Parameter

Creating a Callback Handler

Structure of Callback Handler Data

Cookie
 Major Code
 Minor Code
 Optional Error Code
 HTTP Code

Callback Data Debugging

Optimizing XtraNet Performance

Controlling Idle Time

Using Director's Idle Time Properties

- the frameTempo -or- Tempo Channel
- the cpuHogTicks
- the idleHandlerPeriod
- the idleLoadMode
- the idleLoadPeriod
- the idleReadChunkSize

Sustaining Connections

Credits

Human Code, Inc.

g/matter, inc.

Other g/matter Products

- PrintOMatic Xtra
- PopMenu Xtra
- Sound Xtra
- KILLER Transitions
- Trans-X Xtra
- Director ToolBox Pro
- The MediaBook CD for Director

How To Contact Us

Copyright Notices

Appendix A - Error Messages and Codes

Appendix B - Memory Overhead Chart

Macintosh 68K

Macintosh PPC

Windows 16-bit

Windows 32-bit

Appendix C - Shockwave Security Issues

ISO Compliant Filenames

Invalid Filename Extensions

Appendix D - FTP Connection Process in XtraNet

Appendix E - Additional References and Resources

WWW Resources

WWW References

Book References

- "Internet Protocols Handbook"

Appendix F - URL Encoding

Example #1

Example #2

Overview of XtraNet

XtraNet allows you to add powerful Internet and intranet connectivity to your Macromedia Director productions. For Internet access, XtraNet provides a series of access classes including Server, Client, Peer and Raw. XtraNet also provides direct access to HTTP, FTP and other Internet protocols. The Raw class allows you to create support for custom or special Internet protocols. As an example, a simple SMTP mail application is provided as a sample XtraNet file.

For intranets, XtraNet provides a powerful LAN based multiuser connectivity solution.

You can build a tremendous number of new applications which before now were impossible. Multi-player games, dynamic advertising, multi-user chat systems, incremental news servers, the list of uses is endless.

The Four Types of Xtras

Director 5.0 introduced support for a new type of plug-in which is known as an Xtra. Xtras are available in four different types: Sprite, Lingo, Transition and Tool. XtraNet is a Lingo Xtra.

What is a Sprite Xtra?

A Sprite Xtra (or an Asset Xtra as it is sometimes referred to) adds additional functionality to Director through the introduction of custom castmember types which subsequently can be used to add new types of sprites to your multimedia production's stage.

What is a Lingo Xtra?

A Lingo Xtra adds functionality to Director through the introduction of new commands and functions which can be accessed through Lingo. Lingo Xtras are very similar to XObjects in the way that they add functionality to Director.

What is a Transition Xtra?

A Transition Xtra adds new visual effects functionality to Director to enhance the display of new multimedia elements in a project.

What is a Tool Xtra?

A Tool Xtra adds new functionality to Director through the Xtras menu. Tool Xtras can be written using MOA compliant C or C++ code or you can build a Tool Xtra using Director itself and implementing a movie in a window which is accessed from the Xtras menu.

Version History

1.0

Released 10/16/96
First public release.

1.01

Released 10/31/96

Due to an unfortunate error in the compiled code, the 1.01 release was created to eliminate a timeout that was in the 1.0 version. If you have the 1.0 version you are strongly requested to replace it with the 1.01 release.

1.02

Released 3/1/97

This release adds support for proxy servers using the XNetSetProxy command.

XtraNet Installation and End-user Delivery

How To Install XtraNet with Director

When Director launches, it automatically locates Xtras by looking in two special places:

Macintosh

Location #1: The **Xtras** folder which is in the same location as your Director application.

Location #2: The **Macromedia:Xtras** folder contained in your System folder.

Windows

Location #1: The **Xtras** subdirectory which is in the same location as your Director application.

Location #2: The **Windows\Macromed\Xtras** subdirectory on Windows 3.1 -OR- The **Program Files\Common Files\Macromedia\Xtras** subdirectory on Windows 95 or Windows NT

To make XtraNet available to your copy of Director, you simply need to run the XtraNet Installer contained on the XtraNet CD. Check the XtraNet Read Me file for any late breaking installation news.

Macintosh

Launch the application titled “XtraNet™ Installer” located in the root of the XtraNet CD-ROM.

Windows

Launch the application titled “SETUP.EXE” located in the root of the XtraNet CD-ROM.

How To Install XtraNet with Shockwave

When a Shockwave for Director plug-in launches, it automatically locates Xtras by looking in a special “support” folder. This folder must be located in one of the following places:

Macintosh (All browsers)

68K: The **NP-68K-Dir-Shockwave Folder** folder contained in your browser’s “plug-in” folder.

PowerPC: The **NP-PPC-Dir-Shockwave Folder** folder contained in your browser’s “plug-in” folder.

Windows (Netscape Navigator compatible plug-in)

3.1: The **NP16DSW** directory contained in your browser’s “plug-in” directory.

95/NT: The **NP32DSW** directory contained in your browser’s “plug-in” directory.

Windows (Microsoft ActiveX control)

3.1: The **WINDOWS\SYSTEM** directory.

95/NT: The **WINDOWS\SYSTEM** directory.

To make XtraNet available to your browsers, you run the XtraNet Installer as described above - once for each platform.

Shipping XtraNet with a Runtime Projector

XtraNet can be shipped as part of a Director multimedia project. To do this, you simply need to copy the XtraNet Xtra that you are using in your project into a special folder/subdirectory named **Xtras**. Then you must place this Xtras folder/subdirectory in the same place as your runtime projector application.

If you deliver your project on a CD-ROM, you do not need to install XtraNet onto the user's hard disk unless you require that the user start your program by running the Director runtime projector from the hard disk. If you require this, it is recommended that you create a complete installer for your project which performs the following:

- * Creates any required program groups or folders
- * Checks for the existence of TCP/IP software and if not installed, installs it for the user
- * Copies the Director runtime projector for your project to the hard disk
- * Copies all Director Xtras for your project (including XtraNet) to the hard disk

If you are delivering your content on CD-ROM, but execute it through a Shockwave-compatible browser you will want your installer to place the XtraNet software into the Shockwave support folder (as described above.)

IMPORTANT NOTE: YOU MAY NOT DISTRIBUTE YOUR XTRANET SERIAL NUMBER in any human readable form with your project. The XtraNet serial number is not redistributable under the licensing agreement that accompanies this software and is a copyright violation.

Refer to your Director manual for more detailed instructions on the proper use and delivery of Xtras with your Director projects.

Delivering XtraNet via Shockwave

If you wish to have a user access a webpage and immediately display an interactive multimedia presentation (ostensibly delivered via Shockwave) which also has multiuser and client/server functionality, you do need to install (as mentioned above) the XtraNet software into the user's Shockwave support folder. This installation process can be partially automated through Shockwave and/or other internet protocols, but not completely. The reason for this is security. If webpages were able to automatically place files on the user's harddisk indiscriminately, we would be exposing users to serious potential for "vandalism" by unscrupulous people. To avoid this potential, the user **must** be the arbitrator of the actual placement of the downloaded software into a location which will enable it to function. This protects both the user and the provider of the software.

Here is some simple Lingo code which you can use to determine if XtraNet is installed and if not, direct the user to an HTML document which provides download access and installation instructions. This code would need to be running in a Shockwave movie to function properly:

```
on isXtraNetInstalled
  put the number of xtras into xtraCount
  repeat with theXtraNum = 1 to xtraCount
    if the name of xtra theXtraNum contains "XtraNet" then return TRUE
  end repeat
  return FALSE
end isXtraNetInstalled
```



```
on checkForXtraNet
  if isXtraNetInstalled() = FALSE then
    gotoNetPage "http://www.yoursite.com/getXtraNet.html"
  end if
end checkForXtraNet
```

Getting Started with XtraNet

Using the Xtras Menu

The Tools menu contains an XtraNet item when XtraNet has been properly installed into the Xtras folder. The XtraNet menu command contains two choices:

About XtraNet...

XtraNet Help

About XtraNet...

This command will open up a Director movie in a window which identifies the creators of XtraNet and provides other credits information. It also contains a special "Check XtraNet Version" button which will compare your currently installed version of XtraNet against the current release as identified by our server. This feature requires you have an active connection to the Internet through TCP/IP and that our server is available.

XtraNet Help

This is a help file which contains online documentation for XtraNet in a searchable handy reference which you can access while authoring in Director. It uses the same help engine as Macromedia Director itself so it is easy to use!

Using Lingo

You can determine if XtraNet was successfully loaded when you launched Director by using the "showXlib" Lingo command in the Message window:

```
showXlib
-- XLibraries:
-- Xtra: FileFlex
-- Xtra: XtraNet
-- Xtra: PrintOMatic_Lite
-- "*Standard.xlib"
-- XObject: FileIO      Id:1020
-- XObject: SerialPort  Id:200
-- XObject: XCMDGlue   Id:2020
```

Registering XtraNet

You must apply your serial number to XtraNet using the XNetInit command before distributing XtraNet with your projects. If you enter a valid serial number (usually located on the back of your CD-ROM jewel case) as an argument to the XNetInit command, you will have successfully registered your copy of XtraNet and activated all of its features.

Using an Unregistered Copy of XtraNet

If you don't apply your serial number as described above, you will be reminded that you are using an unregistered version everytime you use the XNetInit command.

Also, using an unregistered version of XtraNet will result in the informational alert dialog appearing every 5 minutes during an XtraNet session.

Finally, two features of XtraNet are disabled when running an unregistered version of XtraNet. The features are the XNetSetConnectionLoc and XNetSetConnectionMsg commands.

The XtraNet Lingo Interface

Overview

XtraNet's features are accessed using Director's Lingo programming language. This documentation assumes that the reader has familiarity with the basic functions and commands provided in Lingo. Below is a completion description of each of the Lingo commands and functions provided by XtraNet.

What is a Function?

A function is a mechanism which you use in your code to evaluate an expression and return a calculated result. You will usually pass arguments to a function although sometimes a function will not require them. You will always expect to get data returned by a function. The data will usually represent the result of evaluating the data passed as arguments to the function itself when it was initially called.

To get a function to perform its work and return a result, you need to surround the argument(s) with a set of parenthesis. Even if the function requires or accepts no arguments, you still need to use the parenthesis to get Director to return the function value. You can also use the special keyword "the result" to get the last function result calculated.

Examples:

```
put myFunction(myData) into calculationResult
put myFunction(myExpression) into calculationResult
put myStaticFunction() into functionResult
```

```
myFunction(myData)
put the result into calculationResult
```

Using functions over properties has an advantage in that functions all return error codes with a zero indicating no error.

What is a Command?

A command is a mechanism which you use in your code to perform a specific task. You may pass a command optional arguments. Generally, you do not expect to get any data returned by a command. If you do, the data will usually be a code which represents the success or failure of the command itself in completing its assigned task.

Examples:

```
executeMyCommand
executeMyCommand myData
put executeMyCommand(myData) into successResult
```

Practically every one of the XtraNet commands returns a status code: 0 for success, and some negative number for an error. You should always check your error codes!

Standard Lingo Xtra Functions/Commands

interface(xtra “XtraNet”) -or- mMessageList(xtra “XtraNet”)

This function returns a short description of all of the functions and commands supported by XtraNet.

```

put interface(xtra “XtraNet”)
-- XtraNet version 1.02
-- Engineered by Brian Sharon
-- Entire program code (C) 1996 Human Code, Inc.
-- XtraNet (C) 1996 g/matter, inc.
-- XtraNet is a registered trademark of Human Code, Inc. under license to g/matter, inc.
-- All rights reserved.
--
-- For more information on XtraNet, see these websites:
-- http://www.humancode.com
-- http://www.gmatter.com
--
-- For technical support and sales, please email:
-- <support@gmatter.com>
-- <sales@gmatter.com>
--
-- Online help is available under the Xtras menu.
--
* XNetVersion
* XNetInit *
* XNetShutdown
* XNetSetProxy *
* XNetIPAddress
* XNetDNSLookup string host
* XNetSetDefaultTimeout integer timeOut
* XNetGetDefaultTimeout
* XNetSetTimeout integer cookie, integer timeOut
* XNetGetTimeout integer cookie
* XNetSetConnectionLoc any point
* XNetSetConnectionMsg string msg
* XNetError integer error
* XNetGetFile string url, string file, string callback
* XNetGetGlobal string url, string global, string callback
* XNetPutFile string file, string url, string callback
* XNetPutGlobal string global, string url, string callback
* XNetSetCookie string cookie
* XNetClearCookie
* XNetIdle
* XNetClose integer cookie
* XNetPort integer cookie
* XNetPeerCreate
* XNetPeerConnect integer cookie, string host, integer port
* XNetPeerAccept integer cookie, integer port
* XNetPeerGet integer cookie
* XNetPeerSend integer cookie, string text
* XNetClientCreate
* XNetClientConnect integer cookie, string host, integer port
* XNetClientAccept integer cookie, integer port
* XNetClientGet integer cookie
* XNetClientSend integer cookie, string text
* XNetServerStart integer maxClients, integer port, string callback
* XNetServerSend integer cookie, string text, *
* XNetServerDisconnect integer cookie, string name
* XNetRawCreate
* XNetRawConnect integer cookie, string host, integer port

```

- * XNetRawAccept integer cookie, integer port
- * XNetRawGet integer cookie
- * XNetRawSend integer cookie, string text

General

These are global level commands and functions which you will need to use to exercise the functionality of XtraNet. Primarily, these let you start and stop XtraNet and your network connection services.

XNetInit

This command initializes your XtraNet instance and starts up your networking connection. You only need to call this once per “session” to initialize your server or client experience.

This command will initiate any network auto-connections through dialup Slip and PPP. If the user already has established a network connection it will be used.

Example:

```
XNetInit
put XNetInit("XNET101-12345-67890") into resultCode
```

If you pass a serial number to XNetInit and you still get the “unregistered” dialog, your serial number was input incorrectly. Please check the number and correct it in your scripts.

XNetIdle

This command provides XtraNet with specific idle time to process your Server object and Data object networking connections. You will want to call this as often as necessary to keep up with the networking demands of your application. A common location for the XNetIdle call will be your on idle handler or inside a frame script which loops on a frame.

When you call XNetIdle, it checks each of your open network connections to determine if the operation is complete and there is a pending callback request. Each connection is examined in a first-in first-out sequence. If a callback is pending, the callback is performed before checking any remaining open connections for their status. This can potentially result in substantial time being spent inside the XNetIdle call.

Note that this command is only necessary for XtraNet Servers and network operations which perform callbacks into your code. If you are using the Client, Peer or Raw mechanisms, you can control idle time by using the appropriate “Get” command. This provides an XtraNet client developer with substantial control over how much time is spent processing the connection versus processing Lingo code or other client side operations.

Examples:

```
on idle
  XNetIdle
end idle

on exitFrame
  XNetIdle
  go to the frame
end exitFrame
```

See the section titled “Optimizing XtraNet Performance” for more details on the use of the XNetIdle command.

XNetClose integer cookieID

This command stops a specific networking operation which you have previously initiated and closes the connection. This is specifically useful for cancelling Get or Put operations.

Example:

```
put XNetClose(cookieID) into cancelResultCode
```

XNetShutdown

This command stops your networking connection. Your instance of XtraNet will still be available and you can re-initialize your networking connection using XNetInit.

Example:

```
on stopMovie
  XNetShutdown
end stopMovie
```

XNetSetProxy string hostName, integer portAddress

This command allows you to set up a proxy server. To set up the proxy server, call XNetSetProxy and provide the hostname and port address of the server. XNetSetProxy provides support for CERN proxies only. To clear the proxy server, pass EMPTY as the argument

Example:

```
on setProxyServer
  XNetSetProxy("proxy.gmatter.com", 4080) into proxyResult
  return proxyResult
end setProxyServer
```

Informational

These functions provide useful XtraNet specific data to you. You can determine which version of XtraNet is running, the IP and domain names for the current machine and more.

XNetVersion()

This function returns the version of this copy of XtraNet as a string.

Note: You can compare this string against the version identifier located at the following URLs to determine if you have the latest version of XtraNet:

<http://www.gmatter.com/xtras/xtranet/xtranetVersion.html>

Example:

```
put XNetVersion() into xtraNetVersion
put XNetVersion()
-- "1.02"
```

XNetIPAddress()

This function returns the IP address of the machine executing this copy of XtraNet as a string.

Example:

```
put XNetIPAddress() into resultCode
-- "255.255.255.17"
```

XNetPort(integer cookieID)

This function returns the port address of the specified network operation (identified by its "cookie") as a number. If the cookieID is invalid, XtraNet will return a value of -1.

Example:

```
put XNetPort(cookieID) into portAddress
```

You can force the system to pick a unused port for you by passing 0 for the port number when using the XNetRawAccept command. You can then find out what the port number was by calling XNetPort with the cookie returned by the XNetRawCreate command.

XNetDNSLookup(string hostname|string IPaddress)

This function returns the IP address of the domain specified by the string hostname.

Failure to resolve to an IP address will return a string of "0.0.0.0"

You can also perform a conversion of an IP address into a domain address.

Failure to resolve to a domain name will return the IP address originally passed in as an argument.

Example:

```
put XNetDNSLookup("206.221.216.1") into hostIP
put hostIP
-- "www.gmatter.com"
```

XNetError(integer errorCode)

This function returns a string description of the error code provided. If an unknown error code is passed in, XtraNet will return "Unknown error". All error codes are negative values except for an error code of 0 which indicates no error. See the Error Code table later in the documentation for more details.

Example:

```
put XNetError(-29) into errorString
alert "An XtraNet error occurred: "&&errorString
```

These commands provide control over the display of the Connecting to Host dialog which is automatically displayed by XtraNet during long connections. The purpose of the dialog is to keep the user informed as to what is occurring when a connection takes longer than usual.

XNetSetConnectionLoc point upperLeftCornerofDialog

This command sets the location of the Connecting to Host dialog which appears after the connection timeout occurs. The dialog will automatically appear after 5 seconds has elapsed and disappear after 60 seconds. This command is not available when using an unregistered version of XtraNet.

Examples:

```
XNetSetConnectionLoc point(0,0)
XNetSetConnectionLoc point(320,240)
```

XtraNet checks to make sure that you have passed a valid point data argument.

Note: You can hide the dialog completely (not recommended) if you pass a point coordinate which is off-screen.

Example:

```
XNetSetConnectionLoc point(-9999,-9999)
```

XNetSetConnectionMsg string dialogText

This command sets the display text of the Connecting to Host dialog which appears after the connection timeout occurs. . The dialog will automatically appear after 5 seconds has elapsed and disappear after 60 seconds. This command is not available when using an unregistered version of XtraNet.

Examples:

```
XNetSetConnectionMsg "Connecting to our super cool server...Please wait a moment!"
XNetSetConnectionMsg field "ConnectionMsg"
```

Note: You can set the display text back to the default by specifying an EMPTY string as the argument.

Example:

```
XNetSetConnectionMsg ""
```

Operation Timing

Use these features to control the automatic timeout mechanisms that will watch over your network connections.

XNetSetDefaultTimeout integer timeOut

This command specifies the default connection timeout value in seconds. All subsequently opened connections will use this timeout value unless the XNetSetTimeout command is used.

Example:

```
-- set the timeout to a full minute due to a slow server connection
XNetSetDefaultTimeout(60)
```

XNetGetDefaultTimeout

This function returns the current default connection timeout value as an integer representing seconds. The default timeout value used in every XtraNet session is 30 seconds. When a timeout occurs, the network operation will return a result value of -29.

Example:

```
put XNetGetDefaultTimeout() into defaultTimeOutLength
```

XNetSetTimeout integer cookie, integer timeOut

This command specifies the connection timeout value in seconds for a specific connection. Specifying a value of 0 will turn off timeouts completely. This is not recommended as it will essentially terminate your network connections as soon as they begin since you will not be allowing for any connection latency. Unlike the XNetSetDefaultTimeout command, you can use this command to set the timeout value for a connection already in progress.

Example:

```
XNetSetTimeout(gCookieID, 15)
```

XNetGetTimeout integer cookie

This function returns the current connection timeout value for a specific connection as an integer representing seconds.

Example:

```
put XNetGetTimeout(gCookieID) into theTimeout
```

Netscape-style Cookies

Use these commands to control Netscape-style cookies that you provide to your XtraNet applications. Basically, a Netscape cookie is a piece of data that an HTTP server can return along with the page you asked for. The server is saying, "hold on to this data, and from now on, whenever you ask me for something, send this data back to me". So a server can use a cookie to do authentication, to track where you've been in a site, to do shopping cart type applications, etc.

XtraNet doesn't really support cookies according to the official specification. If the server returns a cookie, it won't be saved by XtraNet (it's ignored, which is what you're supposed to do if you don't support cookies).

Instead, XtraNet provides a modified form of cookie support which requires that you know the cookie in advance of sending it. You can provide the cookie information to XtraNet by using Javascript in a webpage to get the cookie (`document.cookie`), then we pass that cookie into the Shockwave movie using XtraNet (using parameters to the EMBED tag), and then passing the cookie on to the server from the client using the XNetSetCookie command before making any server requests.

You can read more information about cookies at:

<http://www.graphcomp.com/info/specs/nets/cookie.html>

XNetSetCookie string cookie

This command specifies a string to be sent as the cookie for all future HTTP requests until the XNetClearCookie command is executed or the application terminates. Cookie in this case refers to Netscape's way of implementing persistent client-side states via HTTP, not cookie as the unique ID number returned by many XtraNet calls.

Example:

```
-- set the timeout to a full minute due to a slow server connection  
XNetSetCookie "COOKIENAME=cookiedata"
```

XNetClearCookie

This function returns the current default connection timeout value as an integer representing seconds. The default timeout value used in every XtraNet session is 30 seconds. When a timeout occurs, the network operation will return a result value of -29.

Example:

```
XNetClearCookie
```

HTTP/FTP Access

These commands provide direct access to HTTP and FTP servers to both store and retrieve data. These XtraNet commands automatically determine whether you are perform HTTP or FTP operations and handle the differences for you invisibly. When addressing FTP servers, you can pass in username and password data to establish connection to a secure server.

Example:

```
put XNetGetGlobal("ftp://username:password@www.gmatter.com/private/specialdata.txt",  
"gSpecialData", "processSpecialData") into getCookieID
```

XNetGetFile string url, string filename, string callback

This command gets a remote URL and stores its contents into a local file, using either FTP or HTTP. When the operation finishes, the Lingo handler named "callback" will be notified. See "fetch.dir" for an example.

Example:

```
put XNetGetFile("ftp://ftp.gmatter.com/pub/logo.pic", "gmatter.pic", "importLogo") into getCookieID
```

XNetGetFile can be used to perform CGI GET requests. CGI POST is not currently supported.

Example:

```
put XNetGetFile("http://www.gmatter.com/cgi/mycgi?variable=data", "CGIData", "processCGIresults")  
into getCookieID
```

XNetGetGlobal string url, string globalname, string callback

This command gets a remote URL and stores it into a global variable, using either FTP or HTTP. When the operation finishes, the Lingo handler named "callback" will be notified.

Example:

```
put XNetGetGlobal("http://www.gmatter.com/homepage.html", "gURLData", "HTMLParse") into  
getCookieID
```

XNetGetGlobal can be used to perform CGI GET requests. CGI POST is not currently supported. Under Windows 3.1, you will only be able to retrieve 64K of data into a global variable at one time.

Example:

```
put XNetGetGlobal("http://www.gmatter.com/cgi/mycgi?variable=data", "gCGIData",  
"processCGIresults") into getCookieID
```

XNetPutFile string filename, string url, string callback

This command takes a local file and stores it at a remote URL, using FTP. When the operation finishes, the Lingo handler named "callback" will be notified. See "fetch.dir" for an example.

Example:

put XNetPutFile(the pathname & "userData.txt", "ftp://ftp.gmatter.com/uploads/userdata.txt", "uploadComplete") into putCookieID

XNetPutGlobal string globalname, string url, string callback

This function takes a global variable and stores it at a remote URL, using FTP. When the operation finishes, the Lingo handler named "callback" will be notified. See "fetch.dir" for an example.

Example:

```
put XNetPutGlobal("gFTPSendData", "ftp://ftp.gmatter.com/uploads/userdata.txt", "uploadComplete")
into putCookieID
```

Peer Class

The Peer Class lets you create applications which bypass the unnecessary overhead involved in communicating through an intermediary server. You can even create a "ring" topology by using three or more peers to send direct messages at very fast speeds across a network connection.

The Peer Class is useful for implementing "head to head" experiences or experiences with small groups (2-4).

XNetPeerCreate

This command creates a Peer object that can be used to connect directly to other Peers. "tictac.dir" shows both modes. The comments in that file are pretty extensive as well.

Example:

```
put XNetPeerCreate() into peerObjectCookieID
```

XNetPeerConnect integer cookieID, string remoteHost, integer port

This function connects a Peer object to a specified remote host at the specified port address.

Example:

```
put XNetPeerConnect(peerObjectCookieID, "hostname", 6000) into functionResult
```

XNetPeerAccept integer cookieID, integer port

This command accepts a connection, listening on the specified port. Port numbers should be > 1023 and < 32768 for peer-to-peer and client-server connections.

Example:

```
put XNetPeerAccept(peerObjectCookieID, 6000) into commandResult
```

XNetPeerGet integer cookieID

This function returns a line of text (1K maximum size) received through the specified connection. If there is no data ready to be retrieved, then an empty string ("") is returned. If an error has occurred, an error code will be returned.

Example:

```
put XNetPeerGet(peerObjectCookieID) into textLine
```

XNetPeerSend integer cookieID, string theText

This command send a line of text (1K maximum size) through the specified connection.

Example:

```
put XNetPeerSend(peerObjectCookieID, theText) into commandResult
```

Client Class

The Client Class lets you create client applications which use an intermediary server to communicate. The server acts to collate information from individual clients and then rebroadcast the necessary data to some or all of the clients.

The Client Class is particularly useful for creating multi-player game experiences.

XNetClientCreate

This command creates a Client object that can be connected in a client-server architecture. "tictac.dir" shows both modes. The comments in that file are pretty extensive as well.

Example:

```
put XNetClientCreate() into clientObjectCookieID
```

XNetClientConnect integer cookieID, string remoteHost, integer port

This function connects a Client object to a specified remote host at the specified port address.

Example:

```
put XNetClientConnect(serverObjectCookieID, "hostname", 6000) into functionResult
```

XNetClientAccept integer cookieID, integer port

This command accepts a connection, listening on the specified port. Port numbers should be > 1023 and < 32768 for peer-to-peer and client-server connections.

Example:

```
put XNetClientAccept(serverObjectCookieID, 6000) into functionResult
```

XNetClientGet integer cookieID

This function returns a line of text (1K maximum size) received through the specified connection. If there is no data ready to be retrieved, then an empty string ("") is returned. If an error has occurred, an error code will be returned.

Example:

```
put XNetClientGet(serverObjectCookieID) into textLine
```

XNetClientSend integer cookieID, string theText

This command send a line of text (1K maximum size) through the specified connection.

Example:

```
put XNetClientSend(serverObjectCookieID, theText) into functionResult
```


Server Class

The Server Class lets you create server complements for your client applications. The server acts to collate information from individual clients and then rebroadcast the necessary data to some or all of the clients.

The Server Class is particularly useful for creating multi-player game experiences.

XNetServerStart integer maxClients, integer port, string callback

This function starts an XtraNet server that will accept clients. Arguments allow you to specify how many maximum connections will be allowed and which handler to execute as a callback when connection events occur (i.e. connection made, connection terminated, etc.) Calling this function returns the cookieID associated with the specified port.

Once a Server object has been created, the XNetIdle command will need to be called to provide processing time for connection events. See the XNetIdle command for more information.

Example:

```
put XNetServerStart(50, 6000, "serverStarted") into gServerCookieID
```

The number of active clients will be limited by the number of sockets that the machine can handle. XtraNet current has a per XtraNet-based application process socket limit of 64. You can use XtraNet oriented Lingo code to determine your computer's socket limit (which may be less than XtraNet's):

```
...
set err = 0
set socketCount = 0
XNetInit()
repeat while err >= 0
  set err = XNetGetGlobal("http://www.gmatter.com", "gNetText", "put")
  if err > 0 then set socketCount = socketCount + 1
end repeat
XNetShutdown
...
```

XNetServerSend integer cookieID, string theText [, string clientName]

This command sends a line of text to either all clients or the specified list of clients.

If clientName is not specified, the line of text will be sent to all clients. Otherwise you can pass the internal username assigned to a client (which can be recorded when the client first connects by trapping this information in the "serverCallback" handler.) Only the specified client will receive the line of text. If you want to send the text to a subset of all existing clients, you can use a list variable and process each element inside of a repeat loop. Note: see the "TTSERVER.DIR" example for more illustration of this feature in use.

Example:

```
-- this sends the text to ALL clients
XNetServerSend gCookieID, theText

-- this sends the text to a subset of clients
put ["client1", "client3", "client7"] into clientNameList
repeat with theClient in clientNameList
```

```
XNetServerSend gCookieID, theText, theClient  
end repeat
```

XNetServerDisconnect integer cookieID, string clientName

This command disconnects the specified client.

Example:

```
put XNetServerDisconnect(gCookieID, theClient) into disconnectResult
```

Raw Class

The Raw Class lets you use TCP/IP to communicate using nearly any protocol. The data you send or receive is not altered in any way, nor is any data header added to it. For example, you can implement support for SMTP protocol using Lingo and XtraNet's Raw commands (see the SMTP Client example for more details.)

XNetRawCreate

This function creates a Raw connection object and returns the socket id for the object. You need to create a connection object before using any of the other Raw class commands or functions.

Example:

```
set gMailSock = XNetRawCreate()
```

XNetRawConnect integer cookie, string remoteHost, integer port

This command establishes a connection with the remote host. The host can be any type of server.

Example:

```
set mailhost = "gmatter.com" -- hardcoded port address for SMTP
set mailport = 25 -- SMTP default
put XNetRawConnect(gMailSock, mailhost, mailport) into XNRawResult
```

XNetRawAccept integer cookie, integer port

This command accepts a connection, listening on the specified port. Port numbers should be > 1023 and < 32768 for custom connections. Be careful of conflicting with ports which may already be in use.

Example:

```
set mailport = 2025 -- custom port address for a XtraNet-based SMTP mail server
put XNetRawAccept(gMailSock, mailport)
```

XNetRawGet integer cookie

This function returns any pending data (up to 1024 bytes) being sent by a remote host through the network connection identified by the cookie argument. Only the exact data sent by the remote host will be returned.

Example:

```
put string(XNetRawGet(gMailSock)) into receiverResponse
```

XNetRawSend integer cookie, string theText

This command allows you to send a string to the remote host connected through the connection identified by the cookie argument. The string can contain any data up to 1024 bytes. Only the exact data stored in the string is sent to the remote host.

Example:

```
put XNetRawSend(mailSocket, rawData) into XNetRawSendResult
```

XtraNet's Callback Mechanism

The Callback Parameter

Several of XtraNet's commands allow specification of a custom handler to be called back when the operation specified by that command has been completed.

The callback is used to allow XtraNet to send messages back to your application regarding the state of network operations. It is used by the HTTP and FTP classes (which are created when you call XNetGetFile/ XNetGetGlobal or XNetPutFile/ XNetPutGlobal), and by the XtraNet Server class (created with XNetServerStart).

You can specify any Movie Script level handler for the callback.

Warning: Do not try to use any XtraNet commands or functions inside a callback handler. This is not supported and will generate an XtraNet error result. The only exception to this is the XNetError function which can be used at any time. If you need to make a call to an XtraNet command or function based on callback data that has been received, you should set "flags" using global variables and then process those flags in frame scripts.

Creating a Callback Handler

XtraNet operates asynchronously by sending messages into Director to execute a special handler which you identify when initiating an asynchronous operation. This is simply a Movie Script level will enable constant processing of incoming connection information.

You can create as many different callback handlers as you may need in your project. Every asynchronous operation can have a different callback handler or they can all share the same handler. Your application's functionality will be determine the strategy you will want to apply.

Every callback handler is passed a data structure which identifies the status of the network operation. In order to access this data, your callback handler must specify an argument in which Director can hold the callback handler data. Here's an example:

```
on myCallbackHandler callBackData
-- your code here
end myCallbackHandler
```

When you are specifying the callback handler with the XNetServerStart command, keep in mind that most subsequent server operations will be processed through this callback handler.

Structure of Callback Handler Data

Your callback handlers will receive a data variable as and argument from XtraNet that has the following structure:

cookie	the unique identifier returned when the network operation was started
major code	the type of operation (HTTP, FTP, etc.)
minor code	the message (SUCCESS, ERROR, NEW_PLAYER, etc.)

error	the last error code received
optional	different messages can contain other, additional data
http code	special response data from HTTP servers

The message is sent to the callback handler as a property list. As mentioned above, there are five things that are always put into the list:

Cookie

The unique identifier for the network operation (which we've been calling a cookie). The property is **#cookie**, the value is the id number returned from GetURL, PutFile/PutGlobal, or XNetServStart.

Major Code

A number indicating the type of operation. This is referred to as the major code. The property is **#major**, the possible values are:

- 0 (HTTP)
- 1 (FTP)
- 2 (Server)

Other codes may be added later as new protocols are added.

Minor Code

A number indicating the type of message being sent. This is referred to as the minor code. The property is **#minor**, the possible values are:

- 0 XNET_SUCCESS
- 1 XNET_ERROR
- 2 XNET_STARTED
- 3 XNET_DESTROYED
- 4 XNET_CANCELLED
- 5 XNET_TEXT_MSG
- 8 XNET_NEW_PLAYER
- 9 XNET_PLAYER_LEAVE

For HTTP/FTP, the order of messages is like this:

XNET_STARTED - download/upload started successfully
XNET_SUCCESS/XNET_ERROR - either we finished successfully, or we got an error
XNET_DESTROYED - the instance was deleted, socket freed, etc.

If the user cancels the operation, the order will go like this:

XNET_STARTED - download/upload started successfully
XNET_CANCEL - this operation was cancelled by calling XNetClose
XNET_DESTROYED - the instance was deleted, socket freed, etc.

For HTTP or FTP, you're mainly interested in looking for XNET_SUCCESS and XNET_ERROR. If you're using the server class, you're probably interested in a few more messages. Check out the Tic-Tac-Toe server source code to see how it handles messages like XNET_TEXT_MSG, XNET_NEW_PLAYER, etc.

Optional Error Code

The most recent error code. The property is **#error**, the possible values are the error codes listed here: <http://www.humancode.com/xtranet/errors.html> . The value will always be 0, unless you get an XNET_ERROR message. Alert readers may wonder why we bother to include it if it's always zero. I have no good answer to that question.

HTTP Code

The property is **#httpcode**, the possible values are

200 = success

204 = success but no data

other value = HTTP result code

HTTP servers can generate an error and still return actual data due to redirection and other forms of server-side error handling. Therefore you cannot rely on examining the text returned from an HTTP server to determine if the URL specified was appropriately retrieved. Instead, you should examine the **#httpcode** property to see if any value other than 200 was returned. If a different value is returned, you can analyze it to decide what to do next.

Callback Data Debugging

We recommend that you use Director's internal debugger and set a breakpoint inside your callback handler to better observe the data structure passed in by XtraNet. You may also wish to use the variable watcher window.

Another technique you can use is to specify the Lingo command "put" in place of a callback handler name. This will cause the callback list to be passed as an argument to the put command and correspondingly dumped into the Message window as the data arrives. This is mostly useful to help get familiar with the structure and contents of the callback data variable.

Note that the Client, Peer and Raw classes don't use a callback - instead, you just call the appropriate "Get" command to check for data, and you either get a string back (if something's waiting), or you get an empty string. It's a simpler programming model for complicated games (where having one catch-all handler doesn't always work so well). It also has less overhead, and you get more control over how often you check for messages (since the Client, Peer and Raw don't use XNetIdle to do any idle time processing). This makes it more suited towards multi-player games.

Optimizing XtraNet Performance

There are many methods that can be used to optimize the performance of your XtraNet applications. Writing more efficient code, reducing memory management overhead, using more efficient media assets are all examples of ways to optimize. This chapter focuses on some of the more important techniques.

Controlling Idle Time

One of the most important optimization techniques is to focus on how the computer's noticeable "idle" time is spent. Idle time is simply the computer processing time which has not been previously allocated to other tasks. This is time which you can arbitrarily use to instruct the computer to focus on specifically identified tasks. Normally, Director uses idle time to send "idle" event messages to an executing Director movie. The movie in turn will process these event messages in a special handler called "on idle."

The on idle handler is often used to process code which needs to operate rapidly and continuously throughout a project. For example, the cursor position is often tracking during the on idle handler.

Many XtraNet operations are performed as asynchronous operations. These are operations which are started by XtraNet and sent to another computer. While the remote computer is busy processing the requested operation, XtraNet returns control back to Director so that other local operations can continue.

At some point, XtraNet needs to check the status of the asynchronous operation it started to see whether it is complete and what the result is. To do this, XtraNet provides a special command called XNetIdle which provides CPU time to XtraNet to specifically check the status of all of the existing asynchronous operations. Although you can execute the XNetIdle command at any time, one of the best times to do this is when you know that there is available CPU time to spend performing this status checking task. As a result, many developers will choose to call the XNetIdle command from inside the on idle handler.

Example:

```
on idle
  XNetIdle
end idle
```

Certain features in Director influence how frequently the idle event message is sent to a movie by Director and overall how much idle time is made available. Below you will find suggestions as to how these Director features can be used to provide better performance to XtraNet.

Using Director's Idle Time Properties

Director has a number of Lingo properties related to how idle time is processed which can impact XtraNet's performance. Specifically, the properties are:

```
the frameTempo
the cpuHogTicks
the idleHandlerPeriod
the idleLoadMode
the idleLoadPeriod
the idleReadChunkSize
```

Let's review each property and discuss how they can be used in conjunction with XtraNet to gain maximum performance.

the frameTempo -or- Tempo Channel

The frameTempo property determines the frames per second that Director will attempt to limit the movement of the playback to match. This Lingo property is the same as the setting in the Tempo channel of Director's Score interface.

The higher the frames per second setting, the fewer idle event messages that are sent to a movie by Director. This is because Director is attempting to execute the enterFrame handler, image the frame and then execute the exitFrame handler more and more frequently as the tempo setting is increased. The more time Director spends doing this, the less time is available for sending idle event messages to the movie.

So, to maximize the amount of idle messages and idle time that is available while looping on a frame or running the playback head through a series of frame, you should set the Tempo channel or the frameTempo property to a value of 1.

the cpuHogTicks

Set this property to 0 to give background OS tasks more CPU allocation. The default value is 20. Set this property to a higher value to shutdown OS tasks.

This property only applies to Macintosh and Windows 16-bit. Windows 32-bit uses threads and this property only affects the performance of the thread running Director and XtraNet.

the idleHandlerPeriod

To generate the maximum potential number of idle event messages, set this property to a value of 0 (the default). To force more CPU time to be spent performing other operations and to generate the fewest idle event messages, set this property to a value of the maxInteger (2147483647).

the idleLoadMode

To maximize the amount of idle time that can be spend processing the XNetIdle command, set this property to a value of 0. To create more balance between data loading and XNetIdle command execution, set this property to a value of 1, 2 or 3 (the default.)

the idleLoadPeriod

If the idleLoadMode property is set to 0, you do not need to set this property. Otherwise, set this property to a higher value (up to the maxInteger) to force more time to be made available for XNetIdle command execution and less time for idle time data loading.

the idleReadChunkSize

Set this property to a lower number to create more granularity between XNetIdle command execution and idle time data loading. The smaller the data chunk size, the more interruptable the data loading process is. The default chunk size is 32K. You probably don't want to make the chunk size much smaller than 512 or 1,204 bytes.

Sustaining Connections

Normally, when you are using a web browser to access a HTTP server, a connection is opened for each data element being retrieved from the server to deliver a complete HTML document. If the data contained embedded references to images, for example, each image is delivered by first opening a separate connection to the server, having the server feed the data stream for the image, and then having that connection closed.

Opening and closing server connections takes noticeably CPU time. If you are creating an XtraNet application which needs to repeatedly send or receive data from a remote server, you will find that you can get better performance by keeping the connection open on a sustained basis and reusing that connection for future network operations.

Normally, XtraNet will leave a connection open causing a typical XtraNet application to look like this:

```
XNetConnect  
XNetGet or XNetSend  
XNetGet or XNetSend  
XNetGet or XNetSend  
XNetGet or XNetSend  
...  
XNetClose
```

The main reason to take the alternative approach of closing the connection immediately after moving the data through is if you want to avoid using too many connections. The Macintosh and Windows operating systems have limits as to how many connections you can have active simultaneously. XtraNet itself is limited to 64 connections which is the same as the Macintosh OS limit.

If you take this approach to managing your network operations, your code might look like this:

```
XNetConnect  
XNetGet or XNetSend  
XNetClose  
XNetConnect  
XNetGet or XNetSend  
XNetClose  
XNetConnect  
XNetGet or XNetSend  
XNetClose  
....
```

Credits

XtraNet was created by Human Code, Inc. of Austin, Texas, USA and is exclusively published by g/matter, inc. of San Francisco, California, USA.

Human Code, Inc.

XtraNet Contributors

Concept Design
Chipp Walters

Programming and Implementation
Brian Sharon

Prototype Programming
Scott Thompson

Thanks to Human Coders:
Chris Bohnert
Mary Flanagan
Gary Gattis
Carl Shephard
Cindy Taylor
Masayoshi Umeuchi
Lloyd Walker

Special Thanks

Macintosh socket library based on GUSI (C)1992-1996, Matthias Neeracher,
<http://www.iis.ee.ethz.ch/~neeri>.

Tim Jones
Rick Ligas
Macromedia

for their inspiration and encouragement early on.

Get the most recent information about Human Code, Inc. at their website: <http://www.humancode.com>

g/matter, inc.

g/matter, inc. is the leading publisher of information and tools for mastering multimedia. Established in 1992, we have been responsible for the development of over 70 CD-ROM titles. Since 1994 we have been creating a large number of add-on products for Macromedia software, including:

XtraNet Contributors

Examples and Documentation

Terry R. Schussler

Packaging and Art

David Gartner

Thanks to g/matterians:

Lisa Carroll

Nick Cavander

Skip Knierim

David Kraus

Seth Luersen

Mary Lowry

Chris Mikkelsen

Karyn Scott

Stacey Scott

Go to our website for the latest information on all of our multimedia products: <http://www.gmatter.com>

Other g/matter Products

PrintOMatic Xtra

PrintOMatic is the premier printing tool for Director. The PrintOMatic Xtra adds a full set of page-layout, text and graphics printing features to Macromedia Director projects on Macintosh and Windows. PrintOMatic includes commands for drawing styled text, graphic primitives, bitmapped pictures, and snapshots of the Director stage, all under Lingo control

PopMenu Xtra

PopMenu Xtra adds tools for enhancing your Macromedia Director user interfaces, including pop-up and multi-level heirarchical menus.

Sound Xtra

Sound Xtra adds powerful new sound recording and playback control to your Macromedia Director productions. With Sound Xtra, you can control the use of sound using features which are not available in Lingo or with other Xobjects.

KILLER Transitions

Killer Transitions contains special effects which can be used in your Macromedia Director 5.x and Authorware 3.5 productions to add new visual transitions to your display. Each of the Killer Transitions effects has many powerful settings which can be used to alter the effect to create literally thousands of transition results.

Trans-X Xtra

Trans-X Xtra lets you create your own transition effects by creating animated 1-bit masks and then applying those as a transition. Trans-X ships with over 100 pre-rendered masks to get you started!

Director ToolBox Pro

Director ToolBox Pro contains a powerful suite of development aids for users of Macromedia Director, including cast, score sprite and Lingo analysis, debugging, formatting and management tools which make it easier and faster to produce complete, solid results.

The MediaBook CD for Director

The MediaBook CD for Director is the definitive collection of information and tools for users of Macromedia Director 3, 4 and 5. Learn everything about the Lingo programming language! The MediaBook CD helps you leverage your investment and enrich your use of Director. Professional and novice multimedia producers will save time and money by using the royalty free programming, numerous XObjects for Macintosh and Windows and reuseable, professionally written Lingo provided in The MediaBook CD.

g/matter, inc. also teaches a series of very advanced multimedia development seminars focused on Director, Media Production and Internet development all over the world. Contact us for information on the next seminar near you!

Go to our website for the latest information on all of our multimedia products: <http://www.gmatter.com>

How To Contact Us

If you have technical support questions about using XtraNet, please contact us at:

Email: support@gmatter.com

Fax: +1-415-243-0396

Web: <http://www.gmatter.com>

If you have suggestions for future features or other products please send email to g/matter, inc at:

ideas@gmatter.com

or send us a facsimile at: +1.415.243.0396

Copyright Notices

Programming © 1995-1997 Human Code, Inc. All rights reserved.

Documentation and sample movies © 1995-1997 g/matter, inc. All rights reserved.

XtraNet is a registered trademark of Human Code, Inc. under exclusive license to g/matter, inc.

g/matter, MediaBook and The MediaBook CD are registered trademarks and PopMenu Xtra, PrintOMatic Xtra, Sound Xtra, Director ToolBox Pro are trademarks of g/matter, inc. KILLER Transitions is a registered trademark of SharkByte Tools, Inc. SharkByte is a registered trademark of Terry Ranson. Macromedia, Authorware and Director are registered trademarks and Shockwave is a trademark of Macromedia, Inc. Other product names mentioned within this documentation may be trademarks or registered trademarks of other companies.

Appendix A - Error Messages and Codes

This is a list of result (error) code values, along with their definitions. You can use the XNetError function to convert an error code value into a human-readable string for a given error.

<u>Code</u>	<u>Error String</u>	<u>Error Description</u>
0	"Success"	Congratulations. Whatever you were trying to do, you did it right.
-1	"DNS lookup failure"	DNS (Domain Name Service) failure. We were unable to convert a host name (like www.humancode.com) into an address (like 204.96.160.69). This can indicate many things: a misspelled address, a broken network connection, or the DNS at your service provider may actually be down (the last one happens to me more than I'd like!).
-2	"Failure to create socket"	We were unable to allocate a network connection. The usual cause of this error is being out of memory. Try increasing your Director, runtime projector or internet browser memory partition if you're using a Macintosh.
-3	"This socket is not connected"	Something tried to send or receive data over a connection that was already closed. Note that the other end can close the connection without notifying you.
-4	"Error connecting to remote host"	A common error, with many possible causes. Your network connection can be down. And at the remote end, the computer might be off, not connected to the network, or not listening for connections. The best interpretation of this message is "the remote host is not responding".
-5	"Error closing socket"	An informational error, that there was some problem closing a connection. If there was some data waiting to be delivered, it probably didn't make it. But once the close call has been made there's not much you can do, so you might as well ignore this one.
-6	"Error setting socket options"	An error occurred whiling changing socket attributes. Can (theoretically) arise in the connection process. Never seen this one actually happen, don't know what could cause it, but if you get this error, the connection is toast.

- 7 "Error binding socket to given port value" You asked a server, peer or client to start listening on a specific port, and the system was unable to grant that request. The best way to get that error would be to try to listen on a port that another program is already listening on. If you're running a web server, and then you try to start an XtraNet server listening on port 80, you'll get an error because the web server bound to port 80 before you did.

<u>Code</u>	<u>Error String</u>	<u>Error Description</u>
-8	"Socket unable to listen"	We tried to listen for connections on a specified port, but failed. If you get this error, it'll probably be for the same reason that you would get a "socket creation failure" error: out of memory.
-9	"The remote end has closed the connection"	You tried to do something with the connection, but the other end had closed it down (which they're perfectly entitled to do). Note that if the remote end loses their network connection unexpectedly (like if their modem drops the signal), they won't have a chance to properly close the connection, and you won't receive a remote close message. You have to decide when a person is disconnected based on inactivity. Just wanted to emphasize that.
-10	"Error reading from network"	You can get this error when reading from a network socket.
-11	"Error writing to network"	Like a read error, you can get this error when writing to a socket (just about any XtraNet command).
-12	"Bad argument passed to function"	Right now, that means that you passed in a string that couldn't be converted to a global variable name.
-13	"Invalid URL passed to function"	Double-check your spelling. Make sure the path you're referencing really exists. Make sure (for FTP) that you have permission to read or write the specified URL. Note that XtraNet will accept the syntax ftp://user:pass@host for non-anonymous FTP.
-14	"Error opening file"	Couldn't open the file on your hard disk that you're trying to FTP out of or into. Causes: misspelling, not specifying a full path, no permission to read/create/overwrite the specified file.
-15	"The socket is already connected"	You tried to connect somewhere, when that peer object was already connected to another machine. The current connection is preserved. To close a connection and open another, call XNetCancel() before calling XNetPeerConnect().
-16	"Cannot perform requested operation with given protocol"	An attempt to use an unsupported protocol occurred. Returned if you try to use XNetPutFile or XNetPutGlobal with an http URL (since HTTP 1.0

doesn't support uploading, only downloading), or if you specify an unrecognized URL (like mailto: or gopher:). Check your spelling!

<u>Code</u>	<u>Error String</u>	<u>Error Description</u>
-17	"Out of memory"	Pretty self-explanatory. XtraNet allocates some memory (about 1K) dynamically for each connection you create. If you're starting a bunch of HTTP downloads at the same time, try to control how many connections you open at once.
-18	"Network initialization failed"	Returned by XNetInit() if it fail. You can't do anything else with XtraNet if this call fails. Many possible causes - the network can break in many places, as all Internet users know! Check modem cables, network cables, network software settings.
-19	"There is no job with that identifier"	You passed in an invalid identifier when referencing a connection. This code snippet would return this error to you really quick: <pre>set cookie = XtraNetGetURL("http://www.humancode.com", "gNetText", "callback") set cookie = cookie + 1 set err = XtraNetCancel(cookie)</pre> <p>Guard those cookie values with your life!</p>
-20	"Name already taken"	Obsolete.
-21	"The server is full"	You tried to connect to an XtraNet server, but they had no room for you. Don't take it personally.
-22	"The requested operation cannot be performed on this connection"	Job type mismatch. You passed in a valid identifier for some connection, but it doesn't match up with the function you're calling. A code snippet: <pre>-- get a cookie for an HTTP download set cookie = XtraNetGetURL("http://www.humancode.com", "gNetText", "callback") -- then pass that cookie to a peer function set err = XtraNetHCPPeerSendLine(cookie)</pre> <p>Important note for Windows 3.1 developers: you won't get this error - you'll crash instead! The reason is due to the nature of C++ and the limitations of Visual C++ for Windows 3.1.</p>
-23	"Command failed"	Obsolete.
-24	"Connection refused"	You tried to connect to a host, but the host wasn't listening at the port you wanted.

-25	"Insufficient permission for requested operation"	Your HTTP or FTP request required authentication and the server responded by informing you that you did not have the correct permission.
<u>Code</u>	<u>Error String</u>	<u>Error Description</u>
-26	"Server error"	You should not see this error. If you do, please report it to XTRANET@GMATTER.COM. Thank you.
-27	"There is no user with that name"	You tried to use XNetServerSend to send a message to a specific user, but that user doesn't exist.
-28	"XtraNet has not been initialized"	XtraNet not initialized. Returned if you try to start a connection without having called XtraNetInit successfully.
-29	"Timeout waiting for response"	We were waiting for a response from an FTP or HTTP server, but it didn't arrive within the given time. You can use various XtraNet commands to get and set the default timeout, and to get and set timeouts on a individual connection basis.
-30	"Error reading from file"	You can get this error when reading from a local file via the XtraNetPutFile command. You were able to open the file but unable to read from it. Some sort of disk error occurred when you tried.
-31	"Error writing to file"	You can get this error both when writing to a remote file via the XtraNetGetFile command. The disk is probably full or you encountered some sort of disk error.
-32	"Unable to call XtraNet method from an XtraNet callback handler"	You are making calls to XtraNet functions or commands from within a handler which has been specified as a callback handler. This limitation prevents inadvertent recursion and data corruption. You can however call the XNetError function from within the callback handler.
-33	"Message too long (1 kb max)"	You tried to send a text variable which is larger than 1K (1,024 bytes) in length/size.
-34	"Director internal error"	You should not see this error. If you do, please report it to XTRANET@GMATTER.COM. Thank you.

- 35 "File i/o unavailable in Shockwave" You have attempted to use an XtraNet Get command while specifying a filename which is either not ISO 9660 compliant or has a filename extension which is not allowed. See the Shockwave Security Issues section of the documentation for a list of disallowed extensions.
- 36 "Shockwave security error - invalid file type" You attempted to use the Get or Put commands while in Shockwave in a manner which is not supported by XtraNet. See the Appendix titled "XtraNet Security in Shockwave" for more details.

<u>Code</u>	<u>Error String</u>	<u>Error Description</u>
-37	"Feature not available in unregistered XtraNet"	You attempted to use the SetConnectionMsg and SetConnectionLoc commands without registering XtraNet using the XNetInit command. These features are disabled until XtraNet is registered.
-38	"Invalid FTP user name"	You attempted to access an FTP server using a username which is not known by the server.
-39	"Invalid FTP password"	You attempted to access an FTP server using a password which is incorrect.
-40	"Invalid FTP directory"	You attempted to access an FTP server directory which either doesn't exist or you do not have permission to access.
-41	"Invalid FTP filename"	You attempted to access an FTP server file which either doesn't exist or you do not have permission to access.
-42	"Unknown error"	Some generic XtraNet error has occurred.

Appendix B - Memory Overhead Chart

XtraNet consumes a small amount of additional memory. This table gives you a rough idea of how much based on the user's playback system. You can use the freeBytes property in Lingo to further analyze memory consumption over time.

Macintosh 68K

Base XtraNet Memory Usage: About 200K

Per Connection Memory Usage: About 2K

Macintosh PPC

Base XtraNet Memory Usage: About 86K

Per Connection Memory Usage: About 2K

Note that all of the code can be swapped out of memory when virtual memory is enabled.

Windows 16-bit

Base XtraNet Memory Usage: About 97K

Per Connection Memory Usage: About 2K

Windows 32-bit

Base XtraNet Memory Usage: About 180K

Per Connection Memory Usage: About 2K

Appendix C - Shockwave Security Issues

XtraNet has some security limits which are active only when executing XtraNet through a Shockwave movie. These limits are designed to prevent accidental or malicious use while leaving developers free to flexibly create exciting applications. Here's an overview of the security minded limitations in XtraNet:

ISO Compliant Filenames

XtraNet limits you to use of ISO 9660 (8.3) compatible filenames when using the Get commands. This means you cannot use other than alphanumeric characters in your filename references.

Invalid Filename Extensions

XtraNet will refuse to accept filenames with certain extensions when using the Get commands. This prevents the accidental overwriting of system files and other critical files which involve the proper operation of the computer. The extensions which are not allowed are:

.386	.dll	.lnk	.qtc	.tsp
.3gr	.drv	.pdb	.rat	.upd
.acm	.exe	.prf	.reg	.vbx
.acv	.fon	.map	.rsr	.vxd
.bin	.gid	.mfm	.sig	.w95
.cfg	.hlp	.mod	.spd	.x16
.cpl	.inf	.nls	.sym	.x32
.csp	.ini	.ocx	.tlb	
.d32	.kbd	.olb	.trn	
.dbg	.lic	.q32	.tsk	

Appendix D - FTP Connection Process in XtraNet

This is an overview of the process that XtraNet uses which performing an FTP network operation.

- 1) Check for XtraNet initialization.
Can return "Network not initialized"
- 2) Read the arguments passed to the XtraNet command.
Can return "Director error" if there is something wrong with the arguments.
- 3) Parse the URL.
Can return "Bad url"
XtraNet now matches the URL data to the protocol request.
- 4) Create the connection object.
Can return "Out of memory" but memory would have to be really low.
- 5) Open the network connection.
Can return "DNS fail" or "Connect fail"
- 6) Login to the server
Can return "Bad user", "Bad password", or "Server error"
"bad user" if both are bad.
"server error" is variable as to cause
- 7) Switch to binary mode.
Can return "Server error"
- 8) Change to the directory specified in the URL
Can return "Bad directory" or "Server error"

This can occur because an attempt to change the directory on the server was to an out of bounds place (missing permission, doesn't exist)
- 9) Start another socket to receive the data on
Can return "Socket create fail"
The socket is a randomly selected free one.
- 10) Tell the FTP server which socket to connect to on our end
Can return "Server error" because the server can't connect to socket.
- 11) Send the put or get command
Can return "Bad file" or "Server error"
- 12) If we make it past all of these steps, then we return "success" and a valid cookie value.

Appendix E - Additional References and Resources

WWW Resources

Download the latest version of XtraNet, its documentation and examples at:

<http://www.gmatter.com/download/xtranet.html>

View on-line examples of XtraNet in action using Shockwave at:

<http://www.gmatter.com/products/xtranet/xtranetshowcase.html>

WWW References

You can read more information about Cookies at:

<http://www.graphcomp.com/info/specs/nets/cookie.html>

You can read more information about TELNET (RFC 854) at:

<http://www.graphcomp.com/info/rfc/rfc0854.html>

If you're interested in learning how to implement attachments in a mail application using Xtranet, you will want to read the MIME specification (RFC 1521). The specification can be found at:

<http://www.graphcomp.com/info/rfc/rfc1521.html>

Book References

"Internet Protocols Handbook"

by Dave Roberts

Published by Coriolis Group Books

ISBN 1-883577-88-8

Covers a ton of protocols, including SMTP, POP3, GOPHER, NNTP and TELNET and it has a CD-ROM with all of the current status data on each protocol.

Appendix F - URL Encoding

Example #1

```
-- Functions to encode illegal characters in a URL as per RFC 1738
-- See http://www.graphcomp.com/info/rfc/rfc1738.html for details
--
-- Author: Brian Sharon, brians@humancode.com
```

```
on encodeURL url
  set badList = [" ", "<", ">", QUOTE, "{", "}", "|", "\", "^", "[",
  "]", """]
  set newURL = ""
  repeat with i = 1 to the number of chars in url
    set tChar = char i of url
    set tNum = charToNum(tChar)
    if getOne(badList, tChar) or tNum < 32 or tNum = 127 then
      -- this is an unsafe character, and has
      -- to be encoded. Encoding is a percent sign followed by
      -- the number in hex
      set tChar = "%" & numToHex(tNum)
    end if
    set newURL = newURL & tChar
  end repeat
  return newURL
end encodeURL
```

```
on numToHex num
  -- assumes number is between 0 and 255
  set num1 = num/16
  set num2 = num - num1*16
  case num1 of
    10: set num1 = "A"
    11: set num1 = "B"
    12: set num1 = "C"
    13: set num1 = "D"
    14: set num1 = "E"
    15: set num1 = "F"
  otherwise
    set num1 = string(num1)
  end case
  case num2 of
    10: set num2 = "A"
    11: set num2 = "B"
    12: set num2 = "C"
    13: set num2 = "D"
    14: set num2 = "E"
    15: set num2 = "F"
  otherwise
    set num2 = string(num2)
  end case
  return num1 & num2
end numToHex
```

And here's some examples of the code in use:

```
put encodeurl("http://www.humancode.com")
-- "http://www.humancode.com"
put encodeurl("http://www.humancode.com/directory with spaces/[index.html]")
-- "http://www.humancode.com/directory%20with%20spaces/%5Bindex.html%5D"
```

Example #2

```
on encodeURL oldString
--| This encodes the passed text to a format that is good for the
--| location line in any Browser
--| First written: 96-12-09 Patrik Oskarsson <patrik.oskarsson@shine.se>

if length( oldString ) < 1 then exit -- Exit if the string is empty

set newString = ""
repeat with foo = 1 to length( oldString )
  set currChar to char foo of oldString
  case currChar of
    TAB : set currChar = "%09"
    "~" : set currChar = "%7E"
    ">" : set currChar = "%3E"
    "\" : set currChar = "%5C"
    "^" : set currChar = "%5E"
    "{" : set currChar = "%7B"
    "}" : set currChar = "%7D"
    "<" : set currChar = "%3C"
    "[" : set currChar = "%5B"
    "]" : set currChar = "%5D"
    "\" : set currChar = "%60"
    "|" : set currChar = "%7C"
    " " : set currChar = "+"
    "/" : set currChar = "%2F"
    "#" : set currChar = "%23"
    "?" : set currChar = "%3F"
    "%" : set currChar = "%25"
  otherwise
    nothing
  end case
  set newString = newString & currChar
end repeat

return newString

end encodeURL
```