

The New File Transfer Protocols in Anzio version 12.6

What's New in Version 12.6

With the release of Anzio Lite and AnzioWin version 12.6 we are introducing a new user interface and support for additional file transfer protocols.

Prior releases of AnzioWin supported full Kermit and Zmodem file transfer, as well as a basic ASCII transfer (Open and Transmit of a file). In version 12.4 of AnzioWin, we introduced FTP, a standard Internet file transfer protocol, and in version 12.5 added a user interface to our FTP.

With the release of version 12.6, we have further enhanced our file transfer support by adding a graphical user interface to both Kermit and Zmodem and added support for additional protocols, mainly Xmodem, Ymodem and a simple ASCII Send. And we have made these available to users of Anzio Lite as well as AnzioWin.

But what are the differences? Below we briefly describe each protocol and how it works.

ASCII Send

This protocol is designed to simply read an ASCII file (a file containing printable characters) and send those characters to the host as a stream of data with inter-character and inter-line delays.

Xmodem

Xmodem sends ASCII or binary data as 128 byte blocks with checksum block checking. Additional variations supported include XmodemCRC (128 byte blocks with CRC block checking), Xmodem1K (1024 byte blocks with CRC block checking) and Xmodem1KG (streaming Xmodem1K).

Xmodem sends no file name or file size information, simply the data. So it is up to each end to know what file name to work with.

Ymodem

Ymodem sends ASCII or binary data as 1024 byte blocks (similar to Xmodem1K) with batch file support (multiple file and wildcarding). A variation of Ymodem supported includes YmodemG (streaming Ymodem).

Ymodem does support simple file name and file size information sharing, unlike Xmodem.



Zmodem

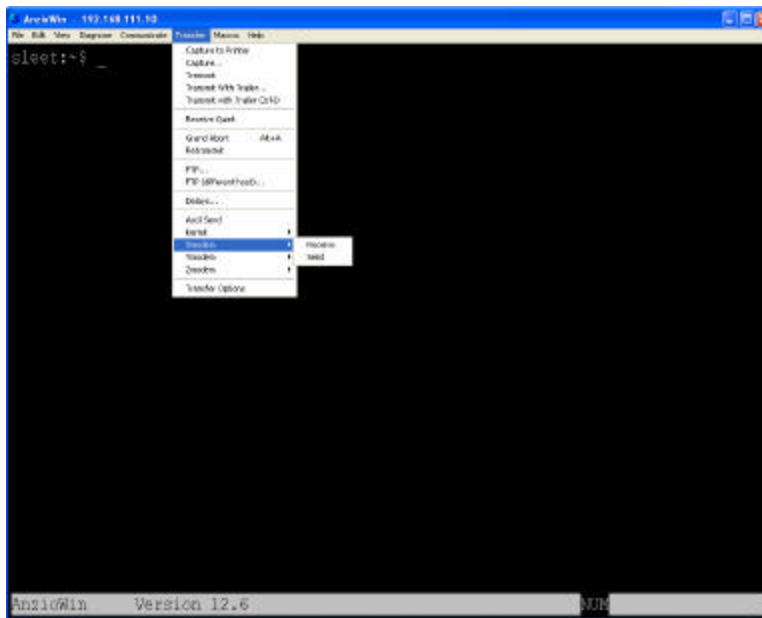
Zmodem sends ASCII or binary data as 1024 byte blocks with batch and streaming capabilities, as well as the ability to restart a file transfer (available in most implementations). Full file name and size and file management options exist.

Kermit

Kermit sends ASCII or binary data as 80 byte blocks with several types of block checking available. Kermit also supports batch, long blocks and windowing (this last improves transfer times dramatically).

The Transfer Menu

Within Anzio Lite and AnzioWin version 12.6, the *Transfer* menu has been expanded to support the latest protocols and the *Transfer Options* dialog menu items. For each transfer, you can elect to either start a Receive or a Send.



Starting A File Transfer

Depending on the protocol selected, you may need to do one of several things before selecting the menu item:

ASCII Send

Before initiating an ASCII send, you need to first start the application on the host you wish to receive the ASCII (printable) characters into. You then select this menu item and the standard Windows Open Dialog will allow you to select a file.

Xmodem Send

Before initiating an Xmodem send, start the host Xmodem receive client, usually "rx <filename>" on UNIX. Remember that Xmodem sends no file information,



File Transfer Protocols In Anzio

you must give the file name with the receive command. Then do this menu item and the standard Windows Open Dialog will allow you to select a file to send.

Xmodem Receive

To initiate an Xmodem receive, first go to the host system and start the Xmodem send client, usually “sx <filename>” on UNIX. When you then do this menu item, the standard Windows SaveAs Dialog will be seen and allow you to select a file name to save the incoming file as (remember that no file information is exchanged in an Xmodem transfer).

Ymodem Send

Before initiating a Ymodem send, start the host Ymodem receive client, usually “rb” on UNIX. Then do this menu item and the standard Windows Open Dialog will allow you to select a file to send.

Ymodem Receive

To initiate a Ymodem receive, first go to the host system and start the Ymodem send client, usually “sb <filename>” on UNIX. When you then do this menu item, the transfer should begin right away.

Zmodem Send

In normal practice, when you start a Zmodem Send, Anzio will try to bring up the “rz” program on a Unix host automatically. If the Zmodem receive program is named “rz” and is available to the user, then you need do nothing at the host. Otherwise, before initiating a Zmodem send, start the host Zmodem receive client. Then do this menu item and the standard Windows Open Dialog will allow you to select a file to send.

Zmodem Receive

To initiate a Zmodem receive, first go to the host system and start the Zmodem send client, usually “sz <filename>” on UNIX. When you then do this menu item, the transfer should begin right away (if auto-receive is turned on, you will not need to start the Zmodem Receive manually in Anzio, see the note at end of this section on Auto-Receive).

Kermit Send

Before initiating a Kermit send, start the host kermit receive client, usually “kermit -r” on UNIX. Then do this menu item and the standard Windows Open Dialog will allow you to select a file to send.

Kermit Receive

To initiate a Kermit receive, first go to the host system and start the kermit send client, usually “kermit -s <filename>” on UNIX. When you then do this menu item, the transfer should begin right away (if auto-receive is turned on, you will not need to start the Kermit Receive manually in Anzio, see the note at end of this section on Auto-Receive).



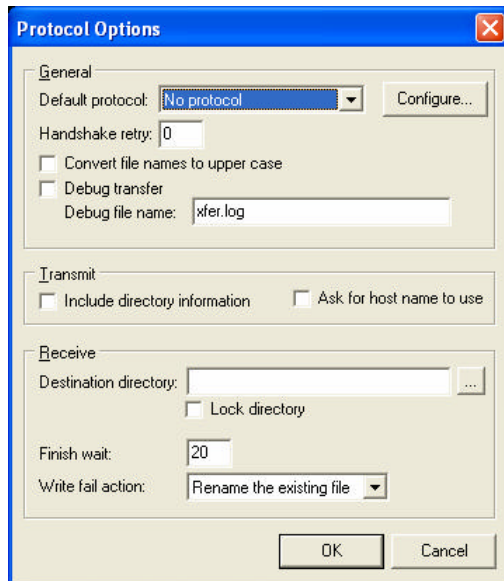
Auto-Receive

Within the Transfer Options dialog (described below) there is an option for Kermit and for Zmodem to turn on auto-reception. When this is enabled, there is no need to do a Kermit Receive, nor a Zmodem Receive, simply do the “kermit -s <filename> or the “sz <filename>” at the host and receiving should automatically start.

Transfer Options

Due to the increase in options available for the various file transfers, we have added a Transfer Options dialog. Through this dialog you can change the Anzio general defaults for file transfers, as well as several protocol values in order to increase efficiency.

First the general values. These apply to all protocols except where the protocol may override the options.



Default Protocol

In order to access the settings for individual protocols, select the protocol from this pull-down list and then click on the Configure... button (see descriptions below for each protocol options).

Handshake retry

This is the number of times to attempt to establish a handshake with the host end before letting a time out occur.

Convert file names to upper case

You can make all names be upper case when sending them to the host or writing them to disk. Protocol options may override this setting.



File Transfer Protocols In Anzio

Debug transfer / Debug file name

When “Debug transfer” is checked, messages between the receiver and sender are written to the “Debug file name”. This file will usually be located in the Anzio directory, or in the Destination directory if “Lock directory” is checked (see below).

Include directory information

When sending a file name to the host (not applicable with ASCII Send or Xmodem Send), we can either include or strip off the current directory information (this may be overridden by the host settings).

Ask for host name to use

When this is checked, prior to sending the file name information to the receiver on a send, the program can prompt the user for the file name to use (this may be overridden by the host settings).

Destination directory

When a file is being received, this is the directory where the file will be saved to (this may be overridden by the host settings in Zmodem and Kermit).

Lock directory

If this item is checked, the “Destination directory” (above) will be used for all files (this will override the host setting if necessary).

Finish wait

Some implementation of certain protocols do not always finish cleanly. In order to prevent the program from simply waiting, we can set a timeout (in milliseconds) to wait (20 is the default).

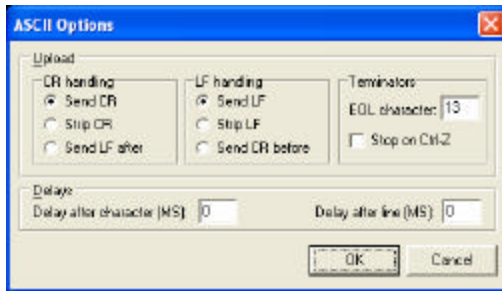
Write fail action

This determines what to do if the file already exists on the PC when receiving a file (this may be overridden by the host). The options are:

- *Fail the attempt*
Skip the file if it already exists
- *Rename the existing file* (default)
We rename the existing file by appending a “\$” to the name. The file name is then used for the incoming file.
- *Overwrite the existing file*
If file already exists, overwrite it
- *Resume*
Attempts to open and append to an already existing file (the protocol must support this option).



ASCII Options



CR handling

This determines how a carriage return will be handled when transferring the ASCII data.

LF handling

This determines how a line feed will be handled when transferring the ASCII data.

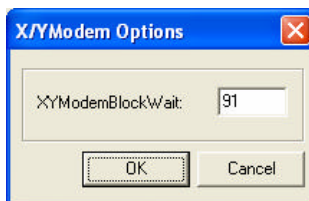
Terminators

This defines the end-of-line character and how to behave on a Ctrl-Z (normal end-of-file).

Delays

You can set the inter-character and inter-line delays to best match what your particular host application needs.

Xmodem / Ymodem Options

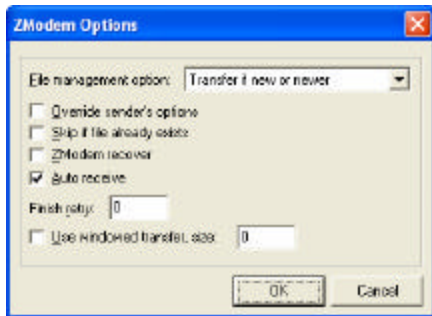


XYModemBlockWait

This is the time to wait for a block of data or a response from the other end of an Xmodem or Ymodem transfer.



Zmodem Options



File management options

This describes how to tell the host to behave if the file already exists at the host end on a file send.

Override sender's options

By enabling this, the host options for file management are overridden and those described in the main Transfer Options dialog (above) are used. It is necessary to set this option in order for the "Write fail action" to work with Zmodem.

Skip if file already exists

This will skip transferring the file if it already exists on the host.

Zmodem recover

When checked, the Zmodem recover option is enabled and will correspond with the host to attempt to restart a transfer (not supported in all implementations of Zmodem).

Auto receive

When checked and the host begins a Zmodem send, Anzio Lite or AnzioWin will begin the Zmodem receive automatically.

Finish retry

This is the number of times to attempt to cleanly finish a transfer.

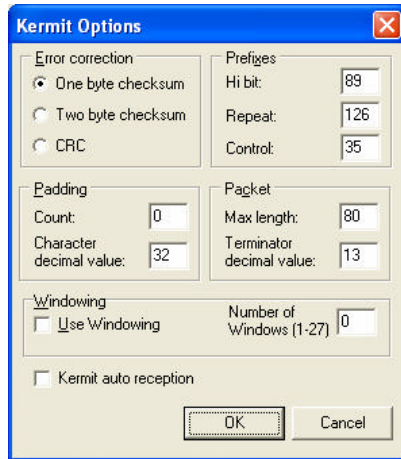
Use windowed transfer, window size

By selecting this item, Zmodem file transfers will wait after the specified number of bytes to let the host system catch up. As an example, with this item checked and a value of 4000, Anzio will wait after every 4000 bytes to let the host catch up.

This option may help solve long file transfers over network links where the communications channel has no flow control.



Kermit Options



Error correction

This is the block checking option to use with transfers.

Prefixes

These are the encoding prefixes to use when encoding non-ascii data. Unless you are familiar with Kermit and the host settings, these should not be changed.

Padding

This is the padding character and count to use when sending data. Unless you are familiar with Kermit and the host settings, these should not be changed.

Packet

This is the maximum block length and the block terminator to use when transferring data. Unless you are familiar with Kermit and the host settings, these should not be changed.

Windowing

This option allows you to enable windowing and set the number of sliding windows to use. By enabling this and setting the number of windows to about 20, you may drastically improve your throughput.

Kermit auto reception

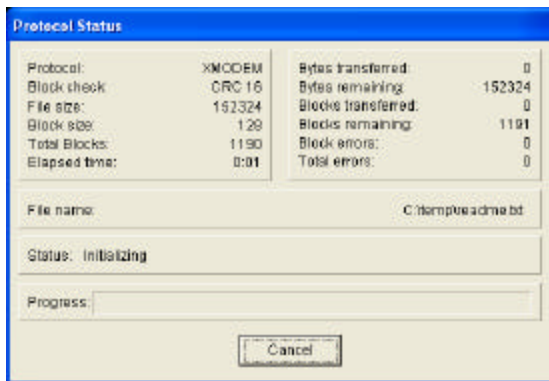
When enabled and you start a Kermit send from the host, Anzio Lite or AnzioWin will automatically start the Kermit Receive.



Protocol Status

We have added a progress status box to all transfers that shows you pertinent information about the transfer in progress (file information and size are subject to the protocol you are using and may not correctly reflect the blocks or the size).

You can press the Cancel button in order to abort a transfer in progress. Depending on the protocol, you may need to take additional steps to abort the transfer. This depends on where you were in the transfer and the amount of queued data (see the notes on Aborting A Transfer below).



Quitting And Aborting Transfers

Various implementations of each protocol can confuse you as to why they did not return you to a UNIX host prompt or back to your program when they completed. Also, sometimes you need to abort a transfer when Anzio thinks it already has. Here are some notes on each protocol and what it takes to quit or abort:

Xmodem

- Receive – need to do extra CR after complete to bring back a prompt

Ymodem

- It takes 2 Ctrl-x to exit Ymodem in general
- On user cancel, it takes a couple of Ctrl-x's

Zmodem

- Receive – need to do extra CR after complete to bring back a prompt
- Sometimes requires several Ctrl-Z (5) plus carriage returns (5) to abort

Aborting a Zmodem file receive that is running over a telnet session (TCP/IP) can sometimes be a chore as there are may be a number of bytes queued up in the communications buffers that are beyond Anzio's control. Caution should be taken here as aborting Zmodem may still cause data to be received - Anzio may think this escaped data is for other purposes, such as passthrough printing.



File Transfer Protocols In Anzio

Kermit

- On user cancel, kermit on host requires a few ctrl-c's to abort.

Scripting A Transfer

From within AnzioWin (not Anzio Lite) you can script most all of the file transfers described here. The general approach is discussed in various documentation on scripting and macros for AnzioWin, available at our web site and in the printed material.

The new command formats are:

asend filename	
ksend [option] filename	Zsend [option] filename
ksend/s [option] filename	Zsend/s [option] filename
Kreceive [option] [filename]	zreceive [option] [filename]
Kreceive/s [option] [filename]	zreceive/s [option] [filename]
Kreceive auto-off / auto-on	zreceive auto-off / auto-on
ysend [option] filename	Xsend [option] filename
ysend/s [option] filename	Xsend/s [option] filename
Yreceive [option] [filename]	xreceive [option] filename
Yreceive/s [option] [filename]	xreceive/s [option] filename

Options

-A	Ascii	-U	Uppercase
-B	Binary	-W xxxx	Windowing and window size
-C	XmodemCRC & Kermit Checksum	-1	KermitCRC
-D	Debug	-2	Kermit Checksum2
-G	Xmodem1KG & YmodemG	-3	Kermit CRC32
-K	Xmodem1K & Kermit CRCK	-8	Zmodem 8K
-O	Zmodem WriteClobber	-V	Debug

Notes:

- Both Kermit and Zmodem support automated reception (no need to run kreceive or zreceive if the auto-receive option is enabled) when a file transfer is initiated from the host.
- You can turn off the auto-reception option of the Kermit and Zmodem protocols by doing “kreceive auto-off” or “zreceive auto-off” from within a script. Likewise, these options can be turned back on with “kreceive auto-on” and “zreceive auto-on”.



File Transfer Protocols In Anzio

- The “/s” option returns a status code from the file transfer, letting the script know of its failure or success. The codes are returned as a 2 character string, “00” through “10”:
 - 00 - successful
 - 01 - at least 1 file tranferred, but at least one was not (not implemented yet)
 - 02 - file not found
 - 03 - file already exists
 - 04 - file skipped
 - 05 - cancelled by user
 - 06 - cancelled by host (not implemented yet)
 - 07 - transfer failed (started ok, but then failed for some reason)
 - 08 - protocol error
 - 09 - protocol timeout (not implemented yet)
 - 10 - critical unknown (not implemented yet)
- When doing an “xreceive”, a filename is required as the protocol does not pass this.
- If you are running Kermit in server mode on the host, you can initiate a file receive at the PC by including the filename as found on the host, i.e. “kreceive filename”. This will do a “get” from the Kermit server.

Notes On This Release

Timings

- Xmodem send is slow.
- Kermit with no options sends at around the same speed as Xmodem.
- Ymodem is approximately twice as fast.
- Kermit with windowing set to 20 is about 30 times faster than Ymodem.
- Zmodem is about 50 times faster than Kermit.
- Zmodem with windowing is about another 10% loss, depending on file size, but may solve problems in long file transfers over networks (it tells Anzio to wait every so often to let the host catch up).

Scripting from the host

Initiating a Zmodem Send from within perl is fairly straight forward. The simplest way to do this is using the “system” command:

```
$bc = "\034";  
$ec = "\035";  
$tmp_dir = "/home/temp";  
system("cd /$tmp_dir && echo -e '$bc zsend -bvy $_[0] $ec' && rz");
```

where \$bc and \$ec are the command brackets for sending commands to Anzio. First start Anzio sending and then start the receiver.



File Transfer Protocols In Anzio

Initiating a Zmodem Receive is the same. You want to start Anzio receiving first, then let the script continue on with the send.

```
$bc = "\034";  
$ec = "\035";  
$tmp_dir = "\\home\\temp";  
system("echo", "-e", " '$bc download-dir $tmp_dir $ec'");  
system("echo -e '$bc zreceive $ec' && sz $_[0]");
```

The routines for starting the host and the PC end, both need to be done together using the && separator, not as separate system commands, or the host script can lose track (the “system” command may not return control back to the script properly).

Scripting From The PC

From within an AnzioWin macro, you could set a macro to do a transfer to the PC:

```
send “kermit -r” | ksend myfilename|
```

or, to go the other direction:

```
kreceive auto-off| send “kermit -s myfilename” | kreceive/s | kreceive auto-on|
```

(this last one would return a status code).

