

Default

COLLABORATORS

	<i>TITLE :</i> Default		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Default	1
1.1	CNet.lib Library Reference	1
1.2	AddFileNote()	3
1.3	AddKeyBuffer()	3
1.4	AddSAM()	3
1.5	AddSAM4()	3
1.6	AppendFile()	4
1.7	CalculateAge()	4
1.8	checkspaces()	4
1.9	ClipboardClose()	5
1.10	ClipboardFreeBuf()	5
1.11	ClipboardOpen()	5
1.12	CBQueryFTXT()	5
1.13	CBReadCHRS()	6
1.14	CBReadDone()	6
1.15	ClipboardReadLong()	6
1.16	ClipboardWriteFTXT()	6
1.17	ClipboardWriteLong()	7
1.18	compstra()	7
1.19	CopyFile()	7
1.20	CopyMessage()	7

Chapter 1

Default

1.1 CNet.lib Library Reference

cnet.lib library reference.

copyright © ZenMetal Software. All rights reserved.

- AddFileNote
- AddKeyBuffer
- AddSAM
- AddSam4
- AppendFile
- CalculateAge
- checkspaces
- ClipboardClose
- ClipboardFreeBuf
- ClipboardOpen
- CBQueryFTEXT
- CBReadCHRS
- CBReadDone
- ClipboardReadLong
- ClipboardWriteFTEXT
- ClipboardWriteLong
- compstra
- CopyFile
- CopyMessage
- CopyText
- DeleteAFile
- DeleteEd
- DelSam
- DelSam4
- DisplayGoKey
- DisposeNUpload
- DoRange
- DumpAll
- DupeSubDirName
- DupeUUCPName
- ExtractFile
- FillClipboardData
- FindArc
- FindPhysNum
- FindSelect

FindUnits
GetDate
GetFileDate
GetFree
GetInfo
GetNext
GetReplyTo
HasMaint
InitNUpload
IsAbort
IsControlLocked
IsNorthAm
IsOnLine
IsPortLoaded
IsPortOccupied
IsPrivate
IsUnvalidated
KillAccount
LoadSelectList
MakeEd
MakeLoc
MakePath
MungeMCI
MyStringSort
NewItem
NoteSelected
NoteTimeTilCutoff
OpenAppend
OpenEd
PrepEditor
PutDateHere
ReadAccount
remMCI
RemoveLeadingSpace
RemoveNonAlNum
RemoveSelect
RemoveSelectLock
SafeFreeMem
SaveAccessData
SaveAccount
SaveAlpha
SaveEditor
SaveNums
SaveReplyTo
SaveSAM
SaveSelectList
SaveSelectListLock
SelectFile
SetTheDoing
SToLower
SToUpper
SuggestUUCPName
TextFromClipboard
TextReload
TextToClipboard
TimeTilCutoff
WriteAllSubboards

```

WriteASubboard
WriteBSubboard
WriteSubboard
WriteUKeys

```

1.2 AddFileNote()

```

Function      : AddFileNote
ProtoType    : void AddFileNote( char *file )
Purpose      : Adds the filenote from z->filenote to the file specified.
               note that the contents of z->myloc are prepended to the ↵
               filename.
Arguments    : file = filename to add z->filenote to.
link requirements: Global pointer to PortData (struct PortData *z)
Returns      : none.
Bugs         : none.

```

1.3 AddKeyBuffer()

```

Function      : AddKeyBuffer
ProtoType    : void AddKeyBuffer( char *string )
Purpose      : Adds the contents of the string to the user's keyboard buffer.
Arguments    : string = the string to add.
link requirements: Global pointer to user's portdata (struct PortData *z)
Returns      : none.
Bugs         : none.

```

1.4 AddSAM()

```

Function      : AddSAM
ProtoType    : void AddSAM( short column, long value )
Purpose      : Increment the specified SAM column by a value.
Arguments    : column = the SAM column index, where 0=the "LAST" column,
               1=the "SETUP" column, 2=the "PERIOD" column,
               3=the "TOTAL" column and 4=the "CURRENT" column.
link requirements: Global pointer to CNet's MainPort data (struct MainPort *myp)
                  Global pointer to CNet PortData (struct PortData *z)
                  Global pointer to CNet's semaphores (struct SignalSemaphore * ↵
                  SEM)
Returns      : none.
Bugs         : none.

```

1.5 AddSAM4()

Function : AddSam4
ProtoType : void AddSAM4(struct MainPort *mainport, short samidx, long ↵
value)
Purpose : Add "value" to the contents of the SAM column 4, row "samidx"
Arguments : samidx = a value representing the row to increment, with
the "feedback" row being samidx=0.
link requirements: none
Returns : none
Bugs : none

1.6 AppendFile()

Function : AppendFile
ProtoType : void AppendFile(char *dest, char *source)
Purpose : To add the contents of one file to the end of another.
Arguments : dest = file to be added to.
source = file containing the text/data to be appended to the
dest file.
link requirements: none
Returns : none
Bugs : none

1.7 CalculateAge()

Function : CalculateAge
ProtoType : short CalculateAge(struct UserData *user)
Purpose : Calculate the age, in years, of the user on the current port.
Arguments : user = pointer to instance of struct UserData containing
a user profile.
link requirements: global pointer to CNet 4 library base (struct Library * ↵
CNet4Base)
Returns : short value representing the user's age in years.
Bugs : none

1.8 checkspaces()

Function : checkspaces
ProtoType : BOOL checkspaces(UBYTE *str, LONG checklen)
Purpose : check a string or portion of a string to see if spaces
exist
Arguments : str = pointer to the string to check for spaces
checklen = the number of characters to check.
link requirements: none
Returns : boolean result. TRUE (1)if spaces were found. FALSE (0)
if there were no spaces in the string.
Bugs : none

1.9 ClipboardClose()

Function : ClipboardClose
ProtoType : void ClipboardClose(struct IOClipReq *ior)
Purpose : Closes the clipboard device and IORequest I/O and ports set up by ClipboardOpen().
Arguments : ior = the same struct IORequest pointer returned by ←
ClipboardOpen().
link requirements: none
Returns : none
Bugs : none

1.10 ClipboardFreeBuf()

Function : ClipboardFreeBuf
ProtoType : void ClipboardFreeBuf(struct cbbuf *buf)
Purpose : Free memory allocated by CReadCHRS().
Arguments : buf = the same buffer pointer returned by CReadCHRS().
link requirements: none
Returns : none
Bugs : none

1.11 ClipboardOpen()

Function : ClipboardOpen
ProtoType : struct IOClipReq *ClipboardOpen(ULONG unit)
Purpose : Open a clipboard unit for read/write.
Arguments : unit = the unit number to open. Limits are 0-255.
0 is the unit most programs use for default clipboard.
CNet VisEd uses units 1-100 for it's user clipboard where
unit 1 is port 0 clipboard, unit 2 is port 1 clipboard etc.
link requirements: none
Returns : returns a pointer to an IOClipReq structure or NULL if
the function fails.
Bugs : none

1.12 CBQueryFTXT()

Function : CBQueryFTXT
ProtoType : int ClipboardQueryFTXT(struct IOClipReq *ior)
Purpose : Check to see if clipboard contains FTXT.
Arguments : ior = a pointer to an IOClipReq structure as returned
by ClipboardOpen().
link requirements: none
Returns : (BOOL) returns TRUE if the clipboard contains an FTXT chunk.
Bugs : none

1.13 CReadCHRS()

Function : CReadCHRS
ProtoType : struct cbbuf *ClipboardReadCHRS(struct IOClipReq *ior)
Purpose : Reads the next CHRS chunk from clipboard
Arguments : ior = a pointer to an IOClipReq structure as returned by ClipboardOpen().
link requirements: none
Returns : pointer to struct cbbuf which MUST be freed using ClipboardFreeBuf() when you are done with the contents.
Bugs : none

1.14 CReadDone()

Function : CReadDone
ProtoType : void ClipboardReadDone(struct IOClipReq *ior)
Purpose : Tell the clipboard we are done reading
Arguments : ior = a pointer to an IOClipReq structure as returned by ClipboardOpen().
link requirements: none
Returns : none
Bugs : none

1.15 ClipboardReadLong()

Function : ClipboardReadLong
ProtoType : int ClipboardReadLong(struct IOClipReq *ior, ULONG *ldata)
Purpose : Read a longword (4 bytes) from the clipboard.
Arguments : ior = a pointer to an IOClipReq structure as returned by ClipboardOpen().
ldata = a pointer to a LONG variable to be filled with the longword read from the clipboard.
link requirements: none
Returns : 1 on success
0 on failure
Bugs : none

1.16 ClipboardWriteFTXT()

Function : ClipboardWriteFTXT
ProtoType : int ClipboardWriteFTXT(struct IOClipReq *ior, char *string)
Purpose : Write a string of text to the clipboard.device
Arguments : ior = a pointer to an IOClipReq structure as returned by ClipboardOpen().
string = a NULL-terminated string to be written to the clipboard.
link requirements: none
Returns : 1 on success
0 on failure
Bugs : none

1.17 ClipboardWriteLong()

Function : ClipboardWriteLong
ProtoType : int ClipboardWriteLong(struct IOClipReq *ior, ULONG *ldata)
Purpose : Write a longword (4 bytes) to the clipboard.
Arguments : ior = a pointer to an IOClipReq structure as returned by ClipboardOpen().
 ldata = a pointer to a LONG variable containing the data to be written to the clipboard.
link requirements: none
Returns : 1 on success
 0 on failure
Bugs : none

1.18 compstra()

Function : compstra
ProtoType : short compstra(char *s, char *t)
Purpose : compare two strings and return result
Arguments : s = the first string to be compared
 t = the second string to be compared
link requirements: none
Returns : 0 if s and t are identical
 >0 if s is alphabetically "higher" than t
 <0 if t is alphabetically "higher" than s
Bugs : none

1.19 CopyFile()

Function : CopyFile
ProtoType : BOOL CopyFile(char *from, char *to)
Purpose : Copy a file.
Arguments : from = the source path and filename to copy.
 to = the destination path and filename.
link requirements: none
Returns : (BOOL) TRUE if successful, FALSE if error occurred. Error information can be obtained from the dos.library IOErr() function.
Bugs : none

1.20 CopyMessage()

Function : CopyMessage
ProtoType : void CopyMessage(BPTR fh, BPTR th)
Purpose : Copy message data from one file to another, stopping at EOF or ^Z. Copying starts at current position of both files and ^Z is written at the end of the text in the destination file.
Arguments : fh = file handle (BPTR) of source file

```

                                th = file handle (BPTR) of destination file.
link requirements: none
Returns                      : none
Bugs                         : none

Function                      : CopyText
ProtoType                    : void CopyText( BPTR fh, BPTR th )
Purpose                      : Copy message data from one file to another, stopping
                                at EOF or ^Z. Copying starts at current position of both
                                files and ^Z is not appended after the text in the destination.
Arguments                    : fh = file handle (BPTR) of source file
                                th = file handle (BPTR) of destination file.
link requirements: none
Returns                      : none
Bugs                         : none

Function                      : DeleteEd
ProtoType                    : void DeleteEd( void )
Purpose                      : Delete the editor file used for the current port
Arguments                    : none
link requirements: none
Returns                      : none
Bugs                         : none

Function                      : DelSAM
ProtoType                    : void DelSAM( short column, long value )
Purpose                      : decrement the specified SAM column by a value.
Arguments                    : column = the SAM column index, where 0=the "LAST" column,
                                1=the "SETUP" column, 2=the "PERIOD" column,
                                3=the "TOTAL" column and 4=the "CURRENT" column.
link requirements: Global pointer to CNet's MainPort data (struct MainPort *myp)
                                Global pointer to CNet PortData (struct PortData *z)
                                Global pointer to CNet's semaphores (struct SignalSemaphore * ←
                                SEM)
Returns                      : none.
Bugs                         : none.

Function                      : DelSam4
ProtoType                    : void DelSAM4( struct MainPort *mainport, short samidx, long ←
                                value )
Purpose                      : Remove "value" from the contents of the SAM column 4, row " ←
                                samidx"
Arguments                    : samidx = a value representing the row to decrement, with
                                the "feedback" row being samidx=0.
link requirements: none
Returns                      : none
Bugs                         : none

Function                      : DisposeNUpload
ProtoType                    : void DisposeNUpload( void )
Purpose                      : Free memory and zero variables used by the ListDir() function
                                of CNet.
Arguments                    : none
link requirements: Global pointer to CNet PortData (struct PortData *z)
                                Global pointer to CNet's semaphores (struct SignalSemaphore * ←
                                SEM)
Returns                      : none

```

```

Bugs                : none

Function            : DoRange
ProtoType           : void DoRange( struct RangeContext *rc, long min, long max, long ←
    defmin, long defmax )
Purpose             : Parse a range of items from z->InBuffer into parts.
                    : Use GetNext to walk the range list once DoRange has parsed them ←
                    :
Arguments           : rc  = pointer to a CNet range context of type (struct ←
    RangeContext)
                    : min = minimum possible range value (lower range boundary).
                    : max = maximum possible range value (upper range boundary).
                    : defmin = minimum range value to be used if there is nothing
                        in z->InBuffer.
                    : defmax = maximum range value to be used if there is nothing
                        in z->InBuffer.
link requirements: Global pointer to CNet PortData (struct PortData *z)
                    : Global pointer to CNetBase (struct Library *CNetBase)
Returns            : fills rc with parsed range data.
Bugs               : none

Function            : DumpAll
ProtoType           : void DumpAll( void )
Purpose             : Logoff all port users immediately
Arguments           : none
link requirements: Global pointer to CNet's MainPort data (struct MainPort *myp)
Returns            : none
Bugs               : none

Function            : DupeSubDirName
ProtoType           : UBYTE DupeSubDirName(char *name, short skipphysnum)
Purpose             : Checks to see if a go keyword is a duplicate while allowing
                    : the go keyword of a certain subboard to be ignored.
Arguments           : name          = the go keyword to be compared with currently
                    :                 configured subboard go keys.
                    : skipphysnum = the physical subboard number belonging to the
                    :                 subboard to be skipped.
link requirements: Global pointer to CNet's MainPort data (struct MainPort *myp)
Returns            : TRUE (1) if existant
                    : FALSE (0) if non-existent
Bugs               : none

Function            : DupeUUCPName
ProtoType           : BOOL DupeUUCPName( char *uucpname, short id )
Purpose             : Checks if Mail id "uucpname" already exists.
Arguments           : uucpname = the mail id to perform dupe checking on.
                    : id       = the account to skip or -1 to check ALL accounts.
link requirements: Global pointer to CNet's MainPort data (struct MainPort *myp)
Returns            : TRUE (1) if mail ID is already used.
                    : FALSE (0) if mail ID is not used.
Bugs               : none

Function            : ExtractFile
ProtoType           : UBYTE ExtractFile( char *name )
Purpose             : View the contents of an archive specified by "name"
Arguments           : name = the full path + filename of the archive to examine.
link requirements: Global pointer to CNet's MainPort data (struct MainPort *myp)

```

```

                                Global pointer to CNet PortData (struct PortData *z)
                                Global pointer to CNetBase (struct Library *CNetBase)
Returns      : TRUE (1) if file extraction success
              : FALSE (0) on failure
Bugs         : none

Function      : FillClipboardData
ProtoType    : struct cbbuf *FillClipboardData( struct IOClipReq *ior, ULONG ←
              size )
Purpose      : Read data from the clipboard into a buffer.
Arguments    : ior = a clipboard request structure as returned by ←
              ClipboardOpen()
              size = length of data to read from the clipboard.
link requirements: none
Returns      : pointer to a cbbuf structure or NULL if error.
Bugs         : none

Function      : FindArc
ProtoType    : short FindArc( char *file )
Purpose      : Get the number of the configured CNet archiver to use for
              file processing.
Arguments    : file = the filename (with extension attached) for which an
              archiver search is required.
link requirements: Global pointer to CNet's MainPort data (struct MainPort *myp)
Returns      : archiver number or 0 on error.
Bugs         : none

Function      : FindPhysNum
ProtoType    : short FindPhysNum(struct SubboardType4 *sp)
Purpose      : Find the physical subboard number for a subboard.
Arguments    : sp = a pointer to an online CNet subboard structure.
link requirements: Global pointer to CNet's MainPort data (struct MainPort *myp)
Returns      : physical subboard number or -1 on error.
Bugs         : none

Function      : FindSelect
ProtoType    : short FindSelect( char *name )
Purpose      : To return the ordinal number of the named select list entry.
Arguments    : name = filename of the select list item.
link requirements: Global pointer to CNet PortData (struct PortData *z)
Returns      : ordinal number of the selection (as the x in z->Select[x])
              or -1 on error/not-found.
Bugs         : none

Function      : FindUnits
ProtoType    : void FindUnits( ULONG size, char *unit, ULONG chunk, char **bm ←
              )
Purpose      : Find the units (k, M, G etc) for a specified bytesize.
Arguments    : size      = the discrete size in BYTES
              unit      = pointer to a string to receive unit information.
              chunk     = the chunk size used for the device.
                      chunk size is only used to calculate byte size if
                      less than 10k.
              bm        = pointer to bbstext lines.
link requirements: none
Returns      : none
Bugs         : none

```

```

Function      : GetDate
ProtoType     : void GetDate( struct IsDate *dest )
Purpose       : Get the current system time in CNet IsDate form.
Arguments     : dest = pointer to an IsDate structure to place the date ←
                information.
link requirements: Global pointer to CNetBase (struct Library *CNetBase)
Returns       : none
Bugs          : Y2k. :) Will be fixed soon.

Function      : GetFileDate
ProtoType     : UBYTE GetFileDate( char *file, struct IsDate *ModDate )
Purpose       : Get file date in CNet IsDate format.
Arguments     : file = path + filename of file for which date information is ←
                desired.
link requirements: Global pointer to CNet PortData (struct PortData *z)
                  Global pointer to CNetBase (struct Library *CNetBase)
Returns       : TRUE (1) if file was found
                  FALSE (0) if file was not found
Bugs          : none

Function      : GetFree
ProtoType     : long GetFree( char *name, UBYTE quiet, char *deststring, char ←
                **bm )
Purpose       : Get freespace of a device/volume
Arguments     : name      = device/volume for which freespace is desired.
                  quiet    = not currently used/OBSOLETE
                  deststring = pointer to string which will receive a size
                              string in bytes, k, M or G.
                  bm       = pointer to BBSTEXT lines.
link requirements: Global pointer to CNetBase (struct Library *CNetBase)
Returns          : size depending on deststring.
                  If deststring is NULL, freespace is returned in BYTES.
                  If deststring is non-NULL, freespace is returned adjusted for
                  k, M, G if required.
Bugs            : none

Function      : GetInfo
ProtoType     : void GetInfo( char *name, char *output )
Purpose       : Get the filenote of a file
Arguments     : name      = the path and filename of the file from which to get
                  the comment.
                  output = pointer to a string that will receive the comment.
link requirements: Global pointer to CNet PortData (struct PortData *z)
Returns          : none
Bugs            : none

Function      : GetNext
ProtoType     : long GetNext( struct RangeContext *rc )
Purpose       : Walk the range list parsed by DoRange
Arguments     : rc = the same RangeContext structure that was initialized/ ←
                filled
                  by DoRange.
link requirements: Global pointer to CNet PortData (struct PortData *z)
Returns          : Next number in the range or -1 on error
Bugs            : none

```

Function : GetReplyTo
ProtoType : UBYTE GetReplyTo(struct UserData *u, char *replytostr)
Purpose : Get the configured reply-to string of the specified user.
Arguments : u = pointer to a UserData structure.
replytostr = pointer to a string that will receive the user's configured reply-to address.

link requirements: none
Returns : TRUE(1) if reply to file exists.
FALSE(0) if reply-to could not be found.
Bugs : none

Function : HasMaint
ProtoType : UBYTE HasMaint(void)
Purpose : Determine if user has maint access either globally or for the current subboard.
Arguments : none
link requirements: Global pointer to CNet PortData (struct PortData *z)
Returns : TRUE (1) if user has either full maint access or SubOp access.
Bugs : none

Function : InitNUpload
ProtoType : UBYTE InitNUpload(void)
Purpose : initialize the upload list. Usually done before ListDir() is used.
Arguments : none
link requirements: Global pointer to CNet's MainPort data (struct MainPort *myp)
Global pointer to CNet PortData (struct PortData *z)
Global pointer to CNet MainPort SignalSemaphores (struct ↵ SignalSemaphore *SEM)
Returns : FALSE(0) on error.
Bugs : none.
