

## EXTENSIONES DEL LENGUAJE Y FUNCIONES

CLIP es un compilador de Clipper, en otras palabras, incluye la sintaxis estándar de Clipper. Debido a esto, consulte los manuales de Clipper para comenzar a programar. Sólo las extensiones del lenguaje serán consideradas aquí.

En la práctica debemos agregar que la sintaxis, está más cercana al Clipper 5.3 (para que lo usen como referencia). Además, aquí sólo se mencionarán estas características extendidas pues no vale la pena su traducción. Si se desea hacer uso de ellas y saber explícitamente como usarlas, debe consultarse el manual oficial de CLIP en inglés, que trata de este tema en el capítulo 10. Si aún no lo obtienes, puedes bajarlo de:

<ftp://ftp.itk.ru/pub/clip/clip-doc-en-html.tgz> en formato HTML

<ftp://ftp.itk.ru/pub/clip/clip-doc-en-pdf.tgz> en formato PDF

### Instrucción SWITCH :

```
[DO] SWITCH <expression>
    CASE <const11>[,<const12>,...]
        <statements>,...
    [ CASE <const21>[,<const22>,...] ]
        <statements>,...
    [ OTHERWISE ]
        <statements>,...
END[SWITCH]
```

### Nuevo operador “:=@” :

```
a := "a" ; b := @a ; ? a,b    // "a","a"
```

### Soporte estilo FoxPro para acceso de arreglos :

```
aVar := {1,2,3,4}
? aVar(nIndex)
```

### Soporte de llamadas a bloques de código vía función :

```
cb := {|x1,x2|qout(x2,x2),"return value"}
? cb(1,2)
```

### Constantes hexadecimales :

```
0xff    //255 en decimal
```

### Aritmética racional :

```
//Compilador convierte a doble precisión, perdiendo precisión  
x := 123456789012345678901234567890
```

```
//Forma correcta de usar constantes de números grandes.  
set(_SET_RATIONAL,.t.) ó SET RATIONAL ON  
x:=val("123456789012345678901234567890")  
x:=val("1234.567890")  
x:=val("1/3")
```

### La función STR ahora soporta los parámetros “len” y “dec” :

```
? str(val("7/3"),1000,990)
```

### Añadida función RSTR que da el resultado en forma de numerador/denominador :

```
? rstr(val("7/3")+val("5/6")) // 19/6
```

### Arreglos asociativos :

```
declare m[5]  
m[1]=11; m[2]=12; ....
```

### Instrucción FOR ... IN :

```
FOR <element> IN <associative_array>  
  <statements>...  
NEXT
```

ó

```
FOR <key> IN <associative_array> KEYS  
  <statements>...  
NEXT
```

### Cadenas como arreglos :

```
string[ <position>, |<length>| ]
```

## Modelo orientado a objetos :

Debido al modelo OO y la compilación a un programa C, existe la posibilidad de escribir las clases estándar Tbrowse y Get en Clipper mismo. Al mismo tiempo, la eficiencia de estas clases NO es peor que aquellas escritas en puro C.

Algunas de las CLASES predefinidas de CLIP, son las siguientes:

COBDEPOSITORY	COBDDICTIONARY	COBIDLIST
CODBLIST	DATETIME	FIND
HISTORY	LISTITEM	MEDIT
QUEUE	SORTEDARRAY	STACK
TEXTEDIT		

Las siguientes CLASES son para uso con la librería gráfica “libclip-ui.so”:

UIBUTTON	UIBUTTONBAR	UICHECKBOX
UICHILDWINDOW	UICHOICE	UICOLOR
UICOMBOBOX	UIDOCUMENT	UIDRIVER
UIEDIT	UIEDITTEXT	UIFONT
UIFORM	UIFRAME	UIGRID
UIHBOX	UIIMAGE	UILABEL
UIMAINWINDOW	UIMENU	UIMENUCHECKEDITEM
UIMENUITEM	UIPOPUPMENU	UISPLITTER
UISTATUSBAR	UITABLE	UITIMER
UITOOLBAR	UITOOLBUTTON	UITREE
UIVBOX	UIWINDOW	UIWORKSPACE

Otras CLASES:

XFL_FORMS	XMLTAG
-----------	--------

## **FUNCIONES CLIP Ordenadas alfabéticamente (no traducidas)**

<a href="#"><u>A2EDIT()</u></a>	Browse 2D array in a table layout
<a href="#"><u>AADD()</u></a>	Add a new element to the end of an array
<a href="#"><u>ABS()</u></a>	Return the absolute value of a numeric expression
<a href="#"><u>ACHOICE()</u></a>	Execute a pop-up menu
<a href="#"><u>ACLONE()</u></a>	Duplicate a nested or multidimensional array
<a href="#"><u>ACOPY()</u></a>	Copy elements from one array to another
<a href="#"><u>ADDSLASHES()</u></a>	Returns string with slashes (\).
<a href="#"><u>ADEL()</u></a>	Delete an array element
<a href="#"><u>ADIR()</u></a>	Fill a series of arrays with directory information

<a href="#"><u>AEVAL()</u></a>	Execute a code block for each element in an array
<a href="#"><u>AFIELDS()</u></a>	Fill arrays with the structure of the current database file
<a href="#"><u>AFILL()</u></a>	Fill an array with a specified value
<a href="#"><u>AINS()</u></a>	Insert a NIL element into an array
<a href="#"><u>ALERT()</u></a>	Display a simple modal dialog box
<a href="#"><u>ALIAS()</u></a>	Return a specified work area alias
<a href="#"><u>ALLTRIM()</u></a>	Remove leading and trailing spaces from a character string
<a href="#"><u>ARRAY()</u></a>	Create an uninitialized array of specified length
<a href="#"><u>ASC()</u></a>	Convert a character to its ASCII value
<a href="#"><u>ASCAN()</u></a>	Scan an array
<a href="#"><u>ASIZE()</u></a>	Grow or shrink an array
<a href="#"><u>ASORT()</u></a>	Sort an array
<a href="#"><u>AT()</u></a>	Return the position of a substring within a character string
<a href="#"><u>ATAIL()</u></a>	Return the highest numbered element of an array
<a href="#"><u>BETWEEN()</u></a>	Check value.
<a href="#"><u>BIN2I()</u></a>	Convert a 16-bit signed integer to a numeric value
<a href="#"><u>BIN2L()</u></a>	Convert a 32-bit signed integer to a numeric value
<a href="#"><u>BIN2W()</u></a>	Convert a 16-bit unsigned integer to a numeric value
<a href="#"><u>BOF()</u></a>	Determine when beginning of file is encountered
<a href="#"><u>BROWSE()</u></a>	Browse records within a window
<a href="#"><u>BUNZIP2()</u></a>	Uncompress string with BZIP2 alghoritm.
<a href="#"><u>BZIP2()</u></a>	Compress string with bZIP2 alghoritm.
<a href="#"><u>BZIP2CLOSE()</u></a>	Close bZIP2 file.
<a href="#"><u>BZIP2OPEN()</u></a>	Open bZip2 file.
<a href="#"><u>BZIP2READ()</u></a>	Read data from bZIP2 file.
<a href="#"><u>BZIP2WRITE()</u></a>	Write data to bZIP2 file.
<a href="#"><u>CDOW()</u></a>	Convert a date value to a character day of the week
<a href="#"><u>CHR()</u></a>	Convert an ASCII code to a character value
<a href="#"><u>CLIP()</u></a>	Evaluate function.
<a href="#"><u>_CLIP_PARND()</u></a>	Retrieve a numeric parameter as a double.
<a href="#"><u>_CLIP_PARNI()</u></a>	Retrieve a numeric parameter as an integer.
<a href="#"><u>_CLIP_PARNL()</u></a>	Retrieve a numeric parameter as a long.
<a href="#"><u>_CLIP_RETND()</u></a>	Post a numeric return value using a double type.
<a href="#"><u>_CLIP_RETNI()</u></a>	Post a numeric return value using an int type.
<a href="#"><u>_CLIP_RETNL()</u></a>	Post a numeric return value using a long type.
<a href="#"><u>_CLIP_STORND()</u></a>	
<a href="#"><u>_CLIP_STORNI()</u></a>	
<a href="#"><u>_CLIP_STORNL()</u></a>	
<a href="#"><u>CMONTH()</u></a>	Convert a date to a character month name
<a href="#"><u>COL()</u></a>	Return the screen cursor column position
<a href="#"><u>COLORSELECT()</u></a>	Activate attribute in current color settings

<a href="#"><u>COM_COUNT()</u></a>	Counts the number of characters in the input buffer.
<a href="#"><u>COM_DTR()</u></a>	Queries/sets the Data Terminal Ready (DTR) status.
<a href="#"><u>COM_FLUSH()</u></a>	Clears the receiving buffer.
<a href="#"><u>COM_HARD()</u></a>	Turns the hardware handshake (automatic CTS) on/off.
<a href="#"><u>COM_INIT()</u></a>	Initializes the port parameters.
<a href="#"><u>COM_NUM()</u></a>	Return the number of the highest available serial interface port
<a href="#"><u>COM_OPEN()</u></a>	Opens the port and initializes the buffer
<a href="#"><u>COMPILEFILE()</u></a>	Compile file.
<a href="#"><u>COMPILESTRING()</u></a>	Compile string.
<a href="#"><u>COM_READ()</u></a>	Reads characters from the receiving buffer.
<a href="#"><u>COM_RTS()</u></a>	Queries/sets the RTS status.
<a href="#"><u>COM_SEND()</u></a>	Transmits data.
<a href="#"><u>COM_SFLUSH()</u></a>	Flush sending buffer.
<a href="#"><u>COM_SOFT()</u></a>	Queries or sets the software handshake (automatic XON/XOFF).
<a href="#"><u>COM_TIMEOUT()</u></a>	Set timeout for all operations in 1/10 seconds.
<a href="#"><u>CSCOUNT()</u></a>	Returns the number specified chars into source string.
<a href="#"><u>CTOD()</u></a>	Convert a date string to a date value
<a href="#"><u>CURDIR()</u></a>	Return the current DOS directory
<a href="#"><u>DATE()</u></a>	Return the system date as a date value
<a href="#"><u>DAY()</u></a>	Return the day of the month as a numeric value
<a href="#"><u>DBAPPEND()</u></a>	Append a new record to the database open in the current work area
<a href="#"><u>DBCLEARFILTER()</u></a>	Clear a filter condition
<a href="#"><u>DBCLEARINDEX()</u></a>	Close all indexes for the current work area
<a href="#"><u>DBCLEARRELATION()</u></a>	Clear active relations
<a href="#"><u>DBCLOSEALL()</u></a>	Close all occupied work areas
<a href="#"><u>DBCLOSEAREA()</u></a>	Close a work area
<a href="#"><u>DBCMMIT()</u></a>	Flush pending updates
<a href="#"><u>DBCMMITALL()</u></a>	Flush pending updates in all work areas
<a href="#"><u>DBCREATE()</u></a>	Create a database file from a database structure array
<a href="#"><u>DBCREATEINDEX()</u></a>	Create an index file
<a href="#"><u>DBDELETE()</u></a>	Mark a record for deletion
<a href="#"><u>DBEDIT()</u></a>	Browse records in a table layout
<a href="#"><u>DBEVAL()</u></a>	Evaluate a code block for each record matching a scope and condition
<a href="#"><u>DBF()</u></a>	Return current alias name
<a href="#"><u>DBFILTER()</u></a>	Return the current filter expression as a character string
<a href="#"><u>DBGOBOTTOM()</u></a>	Move to the last logical record
<a href="#"><u>DBGOTO()</u></a>	Position record pointer to a specific identity
<a href="#"><u>DBGOTOP()</u></a>	Move to the first logical record
<a href="#"><u>DBRECALL()</u></a>	Reinstate a record marked for deletion
<a href="#"><u>DBREINDEX()</u></a>	Recreate all active indexes for the current work area
<a href="#"><u>DBRELATION()</u></a>	Return the linking expression of a specified relation
<a href="#"><u>DBRSELECT()</u></a>	Return the target work area number of a relation

<a href="#"><u>DBSEEK()</u></a>	Move to the record having the specified key value
<a href="#"><u>DBSELECTAREA()</u></a>	Change the current work area
<a href="#"><u>DBSETDRIVER()</u></a>	Return the default database driver and optionally set a new driver
<a href="#"><u>DBSETFILTER()</u></a>	Set a filter condition
<a href="#"><u>DBSETINDEX()</u></a>	Empty orders from an order bag into the order list
<a href="#"><u>DBSETORDER()</u></a>	Set the controlling order
<a href="#"><u>DBSETRELATION()</u></a>	Relate two work areas
<a href="#"><u>DBSKIP()</u></a>	Move relative to the current record
<a href="#"><u>DBSTRUCT()</u></a>	Create an array containing the structure of a database file
<a href="#"><u>DBUNLOCK()</u></a>	Release all locks for the current work area
<a href="#"><u>DBUNLOCKALL()</u></a>	Release all locks for all work areas
<a href="#"><u>DBUSEAREA()</u></a>	Use a database file in a work area
<a href="#"><u>DEFPATH()</u></a>	Returns the true path defined in SET DEFAULT command.
<a href="#"><u>DELETED()</u></a>	Return the deleted status of the current record
<a href="#"><u>DESCEND()</u></a>	Create a descending index key value
<a href="#"><u>DEVOUT()</u></a>	Write a value to the current device
<a href="#"><u>DEVOUTPICT()</u></a>	Write a value to the current device using a picture clause
<a href="#"><u>DEVPOS()</u></a>	Move the cursor or printhead to a new position depending on the current
<a href="#"><u>DIRCHANGE()</u></a>	Change the current DOS directory
<a href="#"><u>DIRECTORY()</u></a>	Create an array of directory and file information
<a href="#"><u>DIRMAKE()</u></a>	Create a directory
<a href="#"><u>DIRREMOVE()</u></a>	Remove a directory
<a href="#"><u>DISKCHANGE()</u></a>	Change the current DOS disk drive
<a href="#"><u>DISKNAME()</u></a>	Return the current DOS drive
<a href="#"><u>DISKSPACE()</u></a>	Return the space available on a specified disk
<a href="#"><u>DISPBEGIN()</u></a>	Begin buffering screen output
<a href="#"><u>DISPBOX()</u></a>	Display a box on the screen
<a href="#"><u>DISPCOUNT()</u></a>	Return the number of pending DISPEND() requests
<a href="#"><u>DISPEND()</u></a>	Display buffered screen updates
<a href="#"><u>DISPOUT()</u></a>	Write a value to the display
<a href="#"><u>DOSERROR()</u></a>	Return the last DOS error number
<a href="#"><u>DOSPATH()</u></a>	Returns file name as DOS file name.
<a href="#"><u>DOW()</u></a>	Convert a date value to a numeric day of the week
<a href="#"><u>DSTRTON()</u></a>	Convert string as "double" to xClipper-numeric.
<a href="#"><u>DTOC()</u></a>	Convert a date value to a character string
<a href="#"><u>DTOS()</u></a>	Convert a date value to a character string formatted as yyymmdd
<a href="#"><u>EMPTY()</u></a>	Determine if the result of an expression is empty
<a href="#"><u>EOF()</u></a>	Determine when end of file is encountered
<a href="#"><u>ERRORBLOCK()</u></a>	Post a code block to execute when a runtime error occurs
<a href="#"><u>ERRORLEVEL()</u></a>	Set the xClipper return code
<a href="#"><u>EVAL()</u></a>	Evaluate a code block
<a href="#"><u>EVP_ALG_LIST()</u></a>	Returns list of algorithms, registered in used version of OpenSSL library.

<a href="#"><u>EVP_DECRYPT()</u></a>	Decrypt string encrypted by ENV_ENCRYPT().
<a href="#"><u>EVP_ENCRYPT()</u></a>	Encrypt string using one of symmetric block algorithm and given key string.
<a href="#"><u>EXP()</u></a>	Calculate e**x
<a href="#"><u>FACCESS()</u></a>	Check access to file.
<a href="#"><u>FCGI_ACCEPT()</u></a>	Accept FCGI connecton from server peer.
<a href="#"><u>FCGI_FLUSH()</u></a>	Flush all buffered data to FCGI server peer.
<a href="#"><u>FCGI_GETENV()</u></a>	Return value of environment variable, or map all variables as name/value pairs.
<a href="#"><u>FCGI_READ()</u></a>	Read output of WWW server, passed via FCGI protocol.
<a href="#"><u>FCLOSE()</u></a>	Close an open binary file and write DOS buffers to disk
<a href="#"><u>FCOUNT()</u></a>	Return the number of fields in the current .dbf file
<a href="#"><u>FCREATE()</u></a>	Create and/or truncate a binary file to zero-length
<a href="#"><u>FERASE()</u></a>	Delete a file from disk
<a href="#"><u>FERROR()</u></a>	Test for errors after a binary file operation
<a href="#"><u>FERRORSTR()</u></a>	Under construction
<a href="#"><u>FIELDBLOCK()</u></a>	Return a set-get code block for a given field
<a href="#"><u>FIELDGET()</u></a>	Retrieve the value of a field using the ordinal position of the field in the
<a href="#"><u>FIELDNAME()</u></a>	Return a field name from the current database (.dbf) file
<a href="#"><u>FIELDPOS()</u></a>	Return the position of a field in a work area
<a href="#"><u>FIELDPUT()</u></a>	Set the value of a field variable using the ordinal position of the field in
<a href="#"><u>FIELDWBLOCK()</u></a>	Return a set-get code block for a field in a given work area
<a href="#"><u>FILE()</u></a>	Determine if files exist in the xClipper default directory or path
<a href="#"><u>FILEATTR()</u></a>	Under construction
<a href="#"><u>FILEATTRS()</u></a>	Returns string - file attributes.
<a href="#"><u>FILEGETSTR()</u></a>	Returns string.
<a href="#"><u>FKLABEL()</u></a>	Return function key name
<a href="#"><u>FKMAX()</u></a>	Return number of function keys as a constant
<a href="#"><u>__FLEDIT()</u></a>	Select the appointed elements from the array of a DbStruct() type.
<a href="#"><u>FLOCK()</u></a>	Lock an open and shared database file
<a href="#"><u>FN_BIN2HEX()</u></a>	Convert a binary string to hexadecimal
<a href="#"><u>FN_BIN2I()</u></a>	Convert a binary string to numeric
<a href="#"><u>FN_CLRBIT()</u></a>	Clear a bit in a number
<a href="#"><u>FN_ERROR()</u></a>	Return current error status for a Netware Library function
<a href="#"><u>FN_GETFSI()</u></a>	Under construction
<a href="#"><u>FN_HEX2BIN()</u></a>	Convert a hexadecimal string to to binary
<a href="#"><u>FN_I2BIN()</u></a>	Convert an integer to binary string
<a href="#"><u>FN_IPXINIT()</u></a>	This function gets the entry address for the IPX interface.
<a href="#"><u>FN_IS3X()</u></a>	Is the current server a 3.x server?
<a href="#"><u>FN_ISBIT()</u></a>	Determine if a specified bit is set
<a href="#"><u>FN_ISNET()</u></a>	Determine if user is on functioning NetWare node
<a href="#"><u>FN_NAMEL()</u></a>	General purpose string to length+string packet function
<a href="#"><u>FN_NETVER()</u></a>	Under construction

<a href="#"><u>FN_NONULL()</u></a>	Strip a string of trailing nulls
<a href="#"><u>FN_PEEKSTR()</u></a>	Read a string of bytes from memory
<a href="#"><u>FN_PFEVAL()</u></a>	Set preferred server and eval block
<a href="#"><u>FN_POKESTR()</u></a>	Write a string of bytes to memory
<a href="#"><u>FN_SETBIT()</u></a>	Set a bit in a number
<a href="#"><u>_FNSETERR()</u></a>	Set the current Netware Library error status
<a href="#"><u>FN_SSFROMP()</u></a>	Remove the file server name from a path string
<a href="#"><u>FN_TOGBIT()</u></a>	Toggle a bit in a number
<a href="#"><u>FOPEN()</u></a>	Open a binary file
<a href="#"><u>FOUND()</u></a>	Determine if the previous search operation succeeded
<a href="#"><u>FREAD()</u></a>	Read characters from a binary file into a buffer variable
<a href="#"><u>FREADSTR()</u></a>	Read characters from a binary file
<a href="#"><u>FRENAME()</u></a>	Change the name of a file
<a href="#"><u>FSEEK()</u></a>	Set a binary file pointer to a new position
<a href="#"><u>FSTRTON()</u></a>	Convert string as "float" to xClipper-numeric.
<a href="#"><u>FT_ADDITION()</u></a>	Add elements unique of source array to target array
<a href="#"><u>FT_AAVG()</u></a>	Average numeric values in an array
<a href="#"><u>FT_ACCTADJ()</u></a>	Adjust beginning or ending fiscal pd. dates to acctg. dates
<a href="#"><u>FT_ACCTMONTH()</u></a>	Return accounting month data
<a href="#"><u>FT_ACCTQTR()</u></a>	Return accounting quarter data
<a href="#"><u>FT_ACCTWEEK()</u></a>	Return accounting week data
<a href="#"><u>FT_ACCTYEAR()</u></a>	Return accounting year data
<a href="#"><u>FT_ACH2TB()</u></a>	Replace ACHOICE() with a Tbrowse object & multiple features.
<a href="#"><u>FT_ADAPTER()</u></a>	Under construction
<a href="#"><u>FT_ADDER()</u></a>	Pop up a simple calculator
<a href="#"><u>FT_ADDWKDY()</u></a>	Return true number of days to add given number of workdays
<a href="#"><u>FT_ADESSORT()</u></a>	Sort an array in descending order
<a href="#"><u>FT_AEMAXLEN()</u></a>	Find longest element within an array
<a href="#"><u>FT_AEMINLEN()</u></a>	Find shortest element within an array
<a href="#"><u>FT_AMEDIAN()</u></a>	Find middle value in array, or average of two middle values
<a href="#"><u>FT_ANOMATCHES()</u></a>	Find the number of array elements meeting a condition
<a href="#"><u>FT_AREDIT()</u></a>	2 dimensional array editing function using TBrowse
<a href="#"><u>FT_ASUM()</u></a>	Sum the elements of an array
<a href="#"><u>FT_AT2()</u></a>	Find position of the nth occurrence of a substring
<a href="#"><u>FT_BITCLR()</u></a>	Clear (reset) selected bit in a byte
<a href="#"><u>FT_BITSET()</u></a>	Set selected bit in a byte
<a href="#"><u>FT_BLINK()</u></a>	Display a blinking message on the screen
<a href="#"><u>FT_BRWSWHL()</u></a>	Browse an indexed database limited to a while condition
<a href="#"><u>FT_BYT2BIT()</u></a>	Convert byte to string of 1's and 0's
<a href="#"><u>FT_BYT2HEX()</u></a>	Convert byte to hexadecimal version of its binary value
<a href="#"><u>FT_BYTEAND()</u></a>	Perform bit-wise AND on two ASCII characters (bytes)
<a href="#"><u>FT_BYTENEG()</u></a>	Perform bit-wise negation on an ASCII character



<a href="#"><u>FT_BYTENOT()</u></a>	Perform bit-wise NOT on an ASCII character (byte)
<a href="#"><u>FT_BYTEOR()</u></a>	Perform bit-wise OR on two ASCII characters (bytes)
<a href="#"><u>FT_BYTEXOR()</u></a>	Perform bit-wise XOR on two ASCII characters (bytes)
<a href="#"><u>FT_CALENDAR()</u></a>	Display date/time calendar, find a date, return calendar data.
<a href="#"><u>FT_CIV2MIL()</u></a>	Convert usual civilian format time to military time.
<a href="#"><u>FT_CLRSEL()</u></a>	User Selectable Colour Routine
<a href="#"><u>FT_D2E()</u></a>	Convert decimal to scientific notation
<a href="#"><u>FT_DATECNFG()</u></a>	Set beginning of year/week for FT_ date functions
<a href="#"><u>FT_DAYOFYR()</u></a>	Return calendar, fiscal or accounting day data
<a href="#"><u>FT_DAYTOBOW()</u></a>	Calculate no. of days between date and beginning of week
<a href="#"><u>FT_DEC2BIN()</u></a>	Convert decimal to binary
<a href="#"><u>FT_DFCLOSE()</u></a>	Close file displayed by FT_DISPFILE()
<a href="#"><u>FT_DFSETUP()</u></a>	Set up parameters for FT_DISPFILE()
<a href="#"><u>FT_DISPFILE()</u></a>	Under construction
<a href="#"><u>FT_DISPMMSG()</u></a>	Display a message and optionally waits for a keypress
<a href="#"><u>FT_DOSVER</u></a>	Return the current DOS major and minor version as a string
<a href="#"><u>FT_DOY()</u></a>	Find number of day within year
<a href="#"><u>FT_DSKFREE()</u></a>	Return the amount of available disk space
<a href="#"><u>FT_DSKSIZE()</u></a>	Return the maximum capacity of a fixed disk
<a href="#"><u>FT_E2D()</u></a>	Convert scientific notation string to a decimal
<a href="#"><u>FT_EASTER()</u></a>	Return the date of Easter
<a href="#"><u>FT_ELAPMIN()</u></a>	Return difference, in minutes, between two mil format times.
<a href="#"><u>FT_ELAPSED()</u></a>	Return elapsed time between two days and/or times
<a href="#"><u>FT_ELTIME()</u></a>	Compute difference between times in hours, minutes, seconds.
<a href="#"><u>FT_ESCCODE()</u></a>	Convert Lotus style escape codes
<a href="#"><u>FT_FAPPEND()</u></a>	Appends a line to the currently selected text file
<a href="#"><u>FT_FBOF()</u></a>	Determine if attempt to skip past beginning of text file
<a href="#"><u>FT_FDAY()</u></a>	Return first day of the month
<a href="#"><u>FT_FDELETE()</u></a>	Deletes a line from the currently selected text file
<a href="#"><u>FT_FEOF()</u></a>	Determine if end of text file has been encountered
<a href="#"><u>FT_FERROR()</u></a>	Return the error code for a text file operation
<a href="#"><u>FT_FGOBOT()</u></a>	Go to the last record in a text file
<a href="#"><u>FT_FGOTO()</u></a>	Move record pointer to specific record in a text file
<a href="#"><u>FT_FGOTOP()</u></a>	Go to the first record in a text file
<a href="#"><u>FT_FILL()</u></a>	Declare menu options for FT_MENU1()
<a href="#"><u>FT_FINDITH()</u></a>	Find the "ith" occurrence of a substring within a string
<a href="#"><u>FT_FINSERT()</u></a>	Inserts a line in the currently selected text file
<a href="#"><u>FT_FLASTRE()</u></a>	Get the no. of records in the currently selected text file
<a href="#"><u>FT_FLOPTST()</u></a>	Test diskette drive status
<a href="#"><u>FT_FREADLN()</u></a>	Read a line from the currently selected text file
<a href="#"><u>FT_FRECNO()</u></a>	Return the current record number of a text file
<a href="#"><u>FT_FSELECT()</u></a>	Select a text file workarea

<a href="#"><u>FT_FSKIP()</u></a>	Move the record pointer to a new position in a text file
<a href="#"><u>FT_FUSE()</u></a>	Open or close a text file for use by the FT_F* functions
<a href="#"><u>FT_FWRITELN()</u></a>	Write a line to the currently selected text file
<a href="#"><u>FT_GCD()</u></a>	Calculate greatest common divisor of two numbers
<a href="#"><u>FT_GETMODE()</u></a>	Get the video mode
<a href="#"><u>FT_GETVCUR()</u></a>	Return info about the cursor on a specified video page
<a href="#"><u>FT_GETVPG()</u></a>	Get the currently selected video page
<a href="#"><u>FT_HEX2DEC()</u></a>	Convert a hex number to decimal
<a href="#"><u>FT_INT86()</u></a>	Under construction
<a href="#"><u>FT_INVCLR()</u></a>	Get the inverse of a color
<a href="#"><u>FT_ISBIT()</u></a>	Test the status of an individual bit
<a href="#"><u>FT_ISBITON()</u></a>	Determine the state of individual bits in a number
<a href="#"><u>FT_ISSHARE()</u></a>	Determine if DOS "Share" is installed
<a href="#"><u>FT_LDAY()</u></a>	Return last day of the month
<a href="#"><u>FT_LINKED()</u></a>	Determine if a function was linked in
<a href="#"><u>FT_MADD()</u></a>	Add or subtract months to/from a date
<a href="#"><u>FT_MBUTPRS()</u></a>	Retrieve button press status
<a href="#"><u>FT_MBUTREL()</u></a>	Get mouse button release information
<a href="#"><u>FT_MCONOFF()</u></a>	Turn mouse cursor off if in specified region
<a href="#"><u>FT_MCURSOR()</u></a>	Set the mouse cursor
<a href="#"><u>FT_MDBLCLK()</u></a>	Return true if a double click was detected
<a href="#"><u>FT_MDEF CRS()</u></a>	Define the mouse cursor
<a href="#"><u>FT_MENU1()</u></a>	Pulldown menu system
<a href="#"><u>FT_MENU2()</u></a>	Vertical lightbar menu
<a href="#"><u>FT_MENUTO()</u></a>	Execute light bar menu using prompts created with @...PROMPT
<a href="#"><u>FT_METAPH()</u></a>	Convert a character string to MetaPhone format
<a href="#"><u>FT_MGETCOORD()</u></a>	Get mouse cursor position (text coord.) and button status
<a href="#"><u>FT_MGETPAGE()</u></a>	Get the display page for the mouse pointer
<a href="#"><u>FT_MGETPOS()</u></a>	Get mouse cursor position and button status
<a href="#"><u>FT_MGETSENS()</u></a>	Get the mouse sensitivity parameters
<a href="#"><u>FT_MGETX()</u></a>	Get mouse cursor row position
<a href="#"><u>FT_MGETY()</u></a>	Get mouse cursor column position
<a href="#"><u>FT_MHIDE CRS()</u></a>	Decrement internal mouse cursor flag and hide mouse cursor
<a href="#"><u>FT_MIL2CIV()</u></a>	Convert time in military format to civilian format.
<a href="#"><u>FT_MIL2MIN()</u></a>	Convert time in military format to number of minute of day.
<a href="#"><u>FT_MIN2DHM()</u></a>	Convert numeric minutes to days, hours and minutes.
<a href="#"><u>FT_MIN2MIL()</u></a>	Convert minute of day to military format time.
<a href="#"><u>FT_MINIT()</u></a>	Initialize the mouse driver, vars and return status of mouse
<a href="#"><u>FT_MINREGION()</u></a>	Test if the mouse cursor is in the passed region
<a href="#"><u>FT_MMICKEYS()</u></a>	Get mickeys
<a href="#"><u>FT_MONTH()</u></a>	Return Calendar or Fiscal Month Data
<a href="#"><u>FT_MRESET()</u></a>	Reset mouse driver and return status of mouse

<a href="#"><u>FT_MSETCOORD()</u></a>	Position the mouse cursor using text screen coordinates
<a href="#"><u>FT_MSETPAGE()</u></a>	Set the display page for the mouse pointer
<a href="#"><u>FT_MSETPOS()</u></a>	Position the mouse cursor using virtual screen coordinates
<a href="#"><u>FT_MSETSENS()</u></a>	Set the mouse sensitivity parameters
<a href="#"><u>FT_MSHOWCRS()</u></a>	Increment internal cursor flag and display mouse cursor
<a href="#"><u>FT_MVERSION()</u></a>	Get the mouse driver version
<a href="#"><u>FT_MXLIMIT()</u></a>	Set vertical bounds of mouse using virtual screen coord.
<a href="#"><u>FT_MYLIMIT()</u></a>	Set horiz. bounds of mouse using virtual screen coordinates
<a href="#"><u>FT_NETPV()</u></a>	Calculate net present value
<a href="#"><u>FT_NOOCCUR()</u></a>	Find the number of times one string occurs in another
<a href="#"><u>FT_NTOW()</u></a>	Translate numeric value to words
<a href="#"><u>FT_NWLSTAT()</u></a>	Return the current Novell NetWare logical station number
<a href="#"><u>FT_NWSEMCLOSE()</u></a>	Close a NetWare semaphore
<a href="#"><u>FT_NWSEMEX()</u></a>	Examine a NetWare semaphore's value and open count
<a href="#"><u>FT_NWSEMLOCK()</u></a>	Perform a semaphore "lock"
<a href="#"><u>FT_NWSEMOPEN()</u></a>	Open or create a NetWare semaphore
<a href="#"><u>FT_NWSEMSIG()</u></a>	Signal a NetWare semaphore (increment)
<a href="#"><u>FT_NWSEMUNLOCK()</u></a>	"Unlock" a semaphore locked by FT_NWSEMLOCK()
<a href="#"><u>FT_NWSEMWAIT()</u></a>	Wait on a NetWare semaphore (decrement)
<a href="#"><u>FT_NWUID()</u></a>	Return the current Novell NetWare User ID
<a href="#"><u>FT_PCHR()</u></a>	Convert printer control codes
<a href="#"><u>FT_PEGS()</u></a>	FT_PEGS GAME (all work and no play...)
<a href="#"><u>FT_PENDING()</u></a>	Display same-line pending messages after a wait.
<a href="#"><u>FT_PICKDAY()</u></a>	Picklist of days of week
<a href="#"><u>FT_POPVID()</u></a>	Restore previously saved video states.
<a href="#"><u>FT_PROMPT()</u></a>	Define a menu item for use with FT_MenuTo()
<a href="#"><u>FT_PUSHVID()</u></a>	Save current video states on internal stack.
<a href="#"><u>FT_QTR()</u></a>	Return Calendar or Fiscal Quarter Data.
<a href="#"><u>FT_RAND1()</u></a>	Generate a random number
<a href="#"><u>FT_RESTARR()</u></a>	Restore a Clipper array from a disc file
<a href="#"><u>FT_RESTSETS()</u></a>	Restore status of all SET command settings
<a href="#"><u>FT_RGNSTACK()</u></a>	Push or pop a saved screen region on or off the stack
<a href="#"><u>FT_ROUND()</u></a>	Rounds a number to a specific place
<a href="#"><u>FT_RSTRGN()</u></a>	Restore region of the screen saved with FT_SAVRGN()
<a href="#"><u>FT_SAVEARR()</u></a>	Save Clipper array to a disc file.
<a href="#"><u>FT_SAVESETS()</u></a>	Save the status of all the SET command settings
<a href="#"><u>FT_SAVRGN()</u></a>	Save a screen region for later display
<a href="#"><u>FT_SCANCODE()</u></a>	Wait for keypress and return keyboard scan code
<a href="#"><u>FT_SETCENTURY()</u></a>	Check/Set the CENTURY Setting
<a href="#"><u>FT_SETDATE()</u></a>	Set the DOS system date
<a href="#"><u>FT_SETMODE()</u></a>	Set the video mode
<a href="#"><u>FT_SETTIME()</u></a>	Set the DOS system time

<a href="#"><u>FT_SETVCUR()</u></a>	Set the cursor position on a specified video page
<a href="#"><u>FT_SETVPG()</u></a>	Set the current video page
<a href="#"><u>FT_SLEEP</u></a>	Wait for a specified amount of time
<a href="#"><u>FT_SQZN()</u></a>	Compress a numeric value into a character string
<a href="#"><u>FT_SYS2MIL()</u></a>	Convert system time to military time format.
<a href="#"><u>FT_SYSMEM()</u></a>	Determine the amount of conventional memory installed
<a href="#"><u>FT_TEMPFIL()</u></a>	Create a file with a unique name
<a href="#"><u>FT_UNSQZN()</u></a>	Uncompress a numeric compressed by FT_SQZN()
<a href="#"><u>FT_WEEK()</u></a>	Return calendar or fiscal week data
<a href="#"><u>FT_WORKDAYS()</u></a>	Return number of work days between two dates
<a href="#"><u>FT_WOY()</u></a>	Find number of week within year
<a href="#"><u>FT_XBOX()</u></a>	Display a self-sizing message box and message
<a href="#"><u>FT_XTOY()</u></a>	Convert from any data type to any other data type
<a href="#"><u>FT_YEAR()</u></a>	Return calendar or fiscal year data
<a href="#"><u>FWRITE()</u></a>	Write to an open binary file
<a href="#"><u>GETACTIVE()</u></a>	Return the currently active Get object
<a href="#"><u>GETAPPLYKEY()</u></a>	Apply a key to a Get object from within a reader
<a href="#"><u>GETCLIPUIVERSION()</u></a>	Get CLIP-UI library version.
<a href="#"><u>GETDOSETKEY()</u></a>	Process SET KEY during GET editing
<a href="#"><u>GETDRIVER()</u></a>	Get current <a href="#"><u>UIDriver</u></a> object.
<a href="#"><u>GETENV()</u></a>	Retrieve the contents of a OS environment variable
<a href="#"><u>GETPOSTVALIDATE()</u></a>	Postvalidate the current Get object
<a href="#"><u>GETPREVALIDATE()</u></a>	Prevalidate a Get object
<a href="#"><u>GETREADER()</u></a>	Execute standard READ behavior for a Get object
<a href="#"><u>GETSYSIMAGE()</u></a>	Get content of predefined icons (XPM format).
<a href="#"><u>GLOB()</u></a>	Check confirmity string to regular expression.
<a href="#"><u>GUNZIP()</u></a>	Uncompress string with GZIP alghoritm.
<a href="#"><u>GZIP()</u></a>	Compress string with GZIP alghoritm.
<a href="#"><u>GZIPCLOSE()</u></a>	Close GZIP file.
<a href="#"><u>GZIPOPEN()</u></a>	Open GZIP file.
<a href="#"><u>GZIPREAD()</u></a>	Read data from GZIP file.
<a href="#"><u>GZIPWRITE()</u></a>	Write data to GZIP file.
<a href="#"><u>HARDCR()</u></a>	Replace all soft carriage returns in a character string with hard carriage
<a href="#"><u>HASHNAME()</u></a>	Returns string from hash values.
<a href="#"><u>HASHSTR()</u></a>	Returns hash code for string.
<a href="#"><u>HEADER()</u></a>	Return the current database file header length
<a href="#"><u>I2BIN()</u></a>	Convert a xClipper numeric to a 16-bit binary integer
<a href="#"><u>IF()</u></a>	Return the result of an expression based on a condition
<a href="#"><u>IIF()</u></a>	Return the result of an expression based on a condition
<a href="#"><u>INDEXEXT()</u></a>	Return the default index extension based on the database driver currently
<a href="#"><u>INDEXKEY()</u></a>	Return the key expression of a specified index
<a href="#"><u>INDEXORD()</u></a>	Return the order position of the controlling index

<a href="#"><u>INKEY()</u></a>	Extract a character from the keyboard buffer or a mouse event
<a href="#"><u>INT()</u></a>	Convert a numeric value to an integer
<a href="#"><u>ISALPHA()</u></a>	Determine if the leftmost character in a string is alphabetic
<a href="#"><u>ISCOLOR()</u></a>	Determine if the current computer has color capability
<a href="#"><u>ISDIGIT()</u></a>	Determine if the leftmost character in a character string is a digit
<a href="#"><u>ISFIELD()</u></a>	Checks if a variable name is a field name of DB.
<a href="#"><u>ISFUNCTION()</u></a>	Checks if a variable name is a function name.
<a href="#"><u>ISLOWER()</u></a>	Determine if the leftmost character is a lowercase letter
<a href="#"><u>ISMEMVAR()</u></a>	Checks if a variable name is a memory variable name.
<a href="#"><u>ISPRINTER()</u></a>	Determine whether LPT1 is ready
<a href="#"><u>ISUPPER()</u></a>	Determine if the leftmost character in a string is uppercase
<a href="#"><u>L2BIN()</u></a>	Convert a xClipper numeric value to a 32-bit binary integer
<a href="#"><u>LASTKEY()</u></a>	Return the INKEY() value of the last key extracted from the keyboard buffer
<a href="#"><u>LASTREC()</u></a>	Determine the number of records in the current .dbf file
<a href="#"><u>LEFT()</u></a>	Extract a substring beginning with the first character in a string
<a href="#"><u>LEN()</u></a>	Return the length of a character string or the number of elements in an array
<a href="#"><u>LIKE()</u></a>	Check confirmity string to simple mask.
<a href="#"><u>LOAD()</u></a>	Load byte-code file or dynamic library.
<a href="#"><u>LOADBLOCK()</u></a>	Load byte-code file.
<a href="#"><u>LOG()</u></a>	Calculate the natural logarithm of a numeric value
<a href="#"><u>LOWER()</u></a>	Convert uppercase characters to lowercase
<a href="#"><u>LTRIM()</u></a>	Remove leading spaces from a character string
<a href="#"><u>LUPDATE()</u></a>	Return the last modification date of a database (.dbf) file
<a href="#"><u>MAKEPATH()</u></a>	Returns path to file in UNIX style.
<a href="#"><u>MAP()</u></a>	Create new empty object (associative array)
<a href="#"><u>MAPGET()</u></a>	Get value from associated array by its key. If specified key is not found, value <defVal> is returned.
<a href="#"><u>MAPKEYS()</u></a>	Under construction
<a href="#"><u>MAPMODIFY</u></a>	Include/exclude control mode for attributes changing.
<a href="#"><u>MAX()</u></a>	Return the larger of two numeric or date values
<a href="#"><u>MAXCOL()</u></a>	Determine the maximum visible screen column
<a href="#"><u>MAXROW()</u></a>	Determine the maximum visible screen row
<a href="#"><u>MCOL()</u></a>	Determine the mouse cursor's screen column position
<a href="#"><u>MDBLCLK()</u></a>	Determine the double-click speed threshold of the mouse
<a href="#"><u>MEMOEDIT()</u></a>	Display or edit character strings and memo fields
<a href="#"><u>MEMOLINE()</u></a>	Extract a line of text from a character string or memo field
<a href="#"><u>MEMOREAD()</u></a>	Return the contents of a disk file as a character string
<a href="#"><u>MEMORY()</u></a>	Determine the amount of available free pool memory
<a href="#"><u>MEMOTRAN()</u></a>	Replace carriage return/linefeeds in character strings
<a href="#"><u>MEMOWRIT()</u></a>	Write a character string or memo field to a disk file
<a href="#"><u>MEMVARBLOCK()</u></a>	Return a set-get code block for a given memory variable
<a href="#"><u>MEMVARGET()</u></a>	Returns MEMVAR value.

<a href="#"><u>MEMVARSET()</u></a>	Sets MEMVAR value.
<a href="#"><u>MENUMODAL()</u></a>	Activate a top bar menu
<a href="#"><u>MHIDE()</u></a>	Hide the mouse pointer
<a href="#"><u>MIN()</u></a>	Return the smaller of two numeric or date values
<a href="#"><u>MLCOUNT()</u></a>	Count the number of lines in a character string or memo field
<a href="#"><u>MLCTOPOS()</u></a>	Return the byte position of a formatted string based on line and column
<a href="#"><u>MLEFTDOWN()</u></a>	Determine the press status of the left mouse button
<a href="#"><u>MLPOS()</u></a>	Determine the position of a line in a character string or memo field
<a href="#"><u>MOD()</u></a>	Return the modulus of two numbers
<a href="#"><u>MONTH()</u></a>	Convert a date value to the number of the month
<a href="#"><u>MPOSTOLC()</u></a>	Return line and column position of a formatted string based on a specified
<a href="#"><u>MPRESENT()</u></a>	Determine if a mouse is present
<a href="#"><u>MRESTSTATE()</u></a>	Re-establish the previous state of a mouse
<a href="#"><u>MRIGHTDOWN()</u></a>	Determine the status of the right mouse button
<a href="#"><u>MROW()</u></a>	Determine a mouse cursor's screen row position
<a href="#"><u>MSAVESTATE()</u></a>	Save the current state of a mouse
<a href="#"><u>MSETBOUNDS()</u></a>	Set screen boundaries for the mouse cursor
<a href="#"><u>MSETCLIP()</u></a>	Define an inclusion region
<a href="#"><u>MSETCURSOR()</u></a>	Determine a mouse's visibility
<a href="#"><u>MSETPOS()</u></a>	Set a new position for the mouse cursor
<a href="#"><u>MSHOW()</u></a>	Display the mouse pointer
<a href="#"><u>MSTATE()</u></a>	Return the current mouse state
<a href="#"><u>NETERR()</u></a>	Determine if a network command has failed
<a href="#"><u>NETNAME()</u></a>	Return the current workstation identification
<a href="#"><u>NEXTKEY()</u></a>	Read the pending key in the keyboard buffer
<a href="#"><u>NOSNOW()</u></a>	Toggle snow suppression
<a href="#"><u>ORDBAGEXT()</u></a>	Return the default order bag RDD extension
<a href="#"><u>ORDBAGNAME()</u></a>	Return the order bag name of a specific order
<a href="#"><u>ORDLISTADD()</u></a>	Add orders to the order list
<a href="#"><u>OS()</u></a>	Return the operating system name
<a href="#"><u>OUTERR()</u></a>	Write a list of values to the standard error device
<a href="#"><u>OUTSTD()</u></a>	Write a list of values to the standard output device
<a href="#"><u>PAD()</u></a>	Pad character, date, and numeric values with a fill character
<a href="#"><u>PARAM()</u></a>	Returns paarameter value.
<a href="#"><u>PCOL()</u></a>	Return the current column position of the printhead
<a href="#"><u>PCOUNT()</u></a>	Determine the position of the last actual parameter passed
<a href="#"><u>PROCLINE()</u></a>	Return the source line number of the current or previous activation
<a href="#"><u>PROCNAME()</u></a>	Return the name of the current or previous procedure or user-defined function
<a href="#"><u>PROW()</u></a>	Return the current row position of the printhead
<a href="#"><u>QOUT()</u></a>	Display a list of expressions to the console
<a href="#"><u>RAT()</u></a>	Return the position of the last occurrence of a substring
<a href="#"><u>READEXIT()</u></a>	Toggle Up arrow and Down arrow as READ exit keys



<a href="#"><u>READFORMAT()</u></a>	Return and optionally, set the code block that implements a format (.fmt)
<a href="#"><u>READINSERT()</u></a>	Toggle the current insert mode for READ and MEMOEDIT()
<a href="#"><u>READKEY()</u></a>	Determine what key terminated a READ
<a href="#"><u>READKILL()</u></a>	Return, and optionally set, whether the current READ should be exited
<a href="#"><u>READMODAL()</u></a>	Activate a full-screen editing mode for a GetList
<a href="#"><u>READUPDATED()</u></a>	Determine whether any GET variables changed during a READ and optionally
<a href="#"><u>READVAR()</u></a>	Return the current GET/MENU variable name
<a href="#"><u>RECCOUNT()</u></a>	Determine the number of records in the current database (.dbf) file
<a href="#"><u>RECNO()</u></a>	Return the identity at the position of the record pointer
<a href="#"><u>RECSIZE()</u></a>	Determine the record length of a database (.dbf) file
<a href="#"><u>REPLICATE()</u></a>	Return a string repeated a specified number of times
<a href="#"><u>RESTSCREEN()</u></a>	Display a saved screen region to a specified location
<a href="#"><u>RIGHT()</u></a>	Return a substring beginning with the rightmost character
<a href="#"><u>RLOCK()</u></a>	Lock the current record in the active work area
<a href="#"><u>ROUND()</u></a>	Return a numeric value rounded to a specified number of digits
<a href="#"><u>ROW()</u></a>	Return the screen row position of the cursor
<a href="#"><u>RSTR()</u></a>	Under construction
<a href="#"><u>RTRIM()</u></a>	Remove trailing spaces from a character string
<a href="#"><u>RUN_BLANK()</u></a>	Under construction
<a href="#"><u>SAVESCREEN()</u></a>	Save a screen region for later display
<a href="#"><u>SCROLL()</u></a>	Scroll a screen region up or down, right or left
<a href="#"><u>SEARCH()</u></a>	Search a substring into string by regular expression.
<a href="#"><u>SECONDS()</u></a>	Return the number of seconds elapsed since midnight
<a href="#"><u>SELECT()</u></a>	Determine the work area number of a specified alias
<a href="#"><u>SET()</u></a>	Inspect or change a system setting
<a href="#"><u>SETBLINK()</u></a>	Toggle asterisk (*) interpretation in the SETCOLOR() string between blinking
<a href="#"><u>SETCANCEL()</u></a>	Toggle Alt+C or Ctrl+C and Ctrl+Break as program termination keys
<a href="#"><u>SETCOLOR()</u></a>	Return the current colors and optionally set new colors
<a href="#"><u>SETCURSOR()</u></a>	Set the cursor shape
<a href="#"><u>SETKEY()</u></a>	Assign an action block to a key
<a href="#"><u>SETMODE()</u></a>	Change display mode to a specified number of rows and columns
<a href="#"><u>SETPOS()</u></a>	Move the cursor to a new position
<a href="#"><u>SETPRC()</u></a>	Set PROW() and PCOL() values
<a href="#"><u>SLEEP()</u></a>	Sleep for the specified number of seconds and milliseconds.
<a href="#"><u>SOUNDEX()</u></a>	Convert a character string to "soundex" form
<a href="#"><u>SPACE()</u></a>	Return a string of spaces
<a href="#"><u>SPLITGEOM()</u></a>	Extract array from string.
<a href="#"><u>SQRT()</u></a>	Return the square root of a positive number
<a href="#"><u>START()</u></a>	Start task.
<a href="#"><u>STARTPATH()</u></a>	Returns full path and name of program.

<a href="#"><u>STR()</u></a>	Convert a numeric expression to a character string
<a href="#"><u>STR2VAR()</u></a>	Convert unicode string to data.
<a href="#"><u>STRTRAN()</u></a>	Search and replace characters within a character string or memo field
<a href="#"><u>STUFF()</u></a>	Delete and insert characters in a string
<a href="#"><u>SUBSTR()</u></a>	Extract a substring from a character string
<a href="#"><u>SYSCMD()</u></a>	Run system command.
<a href="#"><u>TASKGETMSG()</u></a>	Get message.
<a href="#"><u>TASKID()</u></a>	Return task identifier.
<a href="#"><u>TASKPEEKMSG()</u></a>	Get message.
<a href="#"><u>TASKSENDMSG()</u></a>	Send message to task.
<a href="#"><u>TASKSTART()</u></a>	Resume all tasks what was stopped.
<a href="#"><u>TASKSTOP()</u></a>	Stop all tasks with the exclusion current task.
<a href="#"><u>TCPACCEPT()</u></a>	Accept a connection on a socket.
<a href="#"><u>TCPCLOSE()</u></a>	Close connection.
<a href="#"><u>TCPCONNECT()</u></a>	Connect with server.
<a href="#"><u>TCPLISTEN()</u></a>	Listen port.
<a href="#"><u>TCPREAD()</u></a>	Read to buffer data from connection.
<a href="#"><u>TCPWRITE()</u></a>	Commit data from buffer to connection.
<a href="#"><u>TIME()</u></a>	Return the system time
<a href="#"><u>TONE()</u></a>	Sound a speaker tone for a specified frequency and duration
<a href="#"><u>TRANSFORM()</u></a>	Convert any value into a formatted character string
<a href="#"><u>TRIM()</u></a>	Remove trailing spaces from a character string
<a href="#"><u>TYPE()</u></a>	Determine the type of an expression
<a href="#"><u>ULIMIT()</u></a>	Set limit to use system resources.
<a href="#"><u>UPDATED()</u></a>	Determine whether a GET changed during a READ
<a href="#"><u>UPPER()</u></a>	Convert lowercase characters to uppercase
<a href="#"><u>USED()</u></a>	Determine whether a database file is in USE
<a href="#"><u>USEDRIVER()</u></a>	Use specified driver.
<a href="#"><u>VAL()</u></a>	Convert a character number to numeric type
<a href="#"><u>VAL2STR()</u></a>	Transform value to string.
<a href="#"><u>VALTYPE()</u></a>	Determine the data type returned by an expression
<a href="#"><u>VAR2STR()</u></a>	Convert data to unicode string.
<a href="#"><u>VERSION()</u></a>	Returns xClipper version
<a href="#"><u>WEIGHTASC()</u></a>	Returns the weight characteristics of character.
<a href="#"><u>WEIGHTCHR()</u></a>	Returns character by weight characteristics.
<a href="#"><u>WEIGHTTABLE()</u></a>	Returns the weight table of characters.
<a href="#"><u>XMLGETTAG()</u></a>	Get XML element by specified hierarchy.
<a href="#"><u>YEAR()</u></a>	Convert a date value to the year as a numeric value