

Visual LISP Function Reference Chart for AutoCAD 2000/2002/2004

Basic Functions

Arithmetic

(+ [number number] ...)
(- [number number] ...)
(* [number number] ...)
(/ [number number] ...)
(~ int)
(1+ number)
(1- number)
(abs number)
(atan num1 [num2])
(cos ang)
(exp number)
(expt base power)
(fix number)
(float number)
(gcd int1 int2)
(log number)
(logand int int ...)
(logior int int ...)
(lsh int numbers)
(max number number ...)
(min number number ...)
(minusp number)
(rem num1 num2 ...)
(sin ang)

List Manipulation

(sqrt number)
(zerop number)
String-Handling
(read [string])
(strcase string [which])
(strcat string1 string2 ...)
(strlen [string] ...)
(substr string start [length])
(vl-prin1-to-string object)
(vl-princ-to-string object)
(vl-string-elt string position)
(vl-string-left-trim char-set string)
(vl-string-mismatch str1 str2 [pos1 pos2 ignore-case-p])
(vl-string-position char-code str [start-pos [from-end-p]])
(vl-string-right-trim char-set string)
(vl-string-search pattern string [start-pos])
(vl-string-subst new-str pattern str [start-pos])
(vl-string-translate source-set dest-set str)
(vl-string-trim char-set string)
(vl-string->list string)
(wcmatch string pattern)

Equality & Conditional

(= numstr [numstr] ...)
(/= numstr [numstr] ...)
(< numstr [numstr] ...)
(<= numstr [numstr] ...)
(> numstr [numstr] ...)
(>= numstr [numstr] ...)
(and [expr ...])
(Boole func int1 [int2 ...])
(cond [(test result ...) ...])
(eq expr1 expr2)
(equal expr1 expr2 [fuzcz])

(if testexpr thenexpr [elseexpr])
(or [expr ...])
(repeat int [expr ...])
(while testexpr [expr ...])

Error-Handling

(alert string)
(*error* string)
(exit)
(quit)
(vl-catch-all-apply 'function list)
(vl-catch-all-error-message error-obj)
(vl-catch-all-error-p arg)

Symbol-Handling

(atom item)
(atoms-family format [symlist])
(boundp sym)
(not item)
(null item)
(numberp item)
(quote expr)
(set sym expr)
(setq sym1 expr1 [sym2 expr2] ...)
(type item)
(vl-symbol-name symbol)
(vl-symbol-value symbol)
(vl-symbolp object)

List Manipulation

(acad_strlsort list)
(append list ...)
(assoc item alist)
(car list)
(cdr list)
(cons new-first-element list)
(foreach name list [expr ...])
(last list)
(length list)
(list [expr ...])
(listp item)
(mapcar function list1 ... listn)
(member expr list)
(nth n list)
(reverse list)
(subst newitem olditem list)
(vl-consp list-variable)
(vl-every predicate-func list [more-lists] ...)
(vl-list* object [more-objects] ...)
(vl-list->string char-codes-list)
(vl-list-length list-or-cons-object)
(vl-member-if predicate-function list)
(vl-member-if-not predicate-func list)
(vl-position symbol list)
(vl-remove element-to-remove list)
(vl-remove-if predicate-func list)
(vl-remove-if-not predicate-func list)
(vl-some predicate-func list [more-lists] ...)
(vl-sort list less?-func)
(vl-sort-i list less?-func)
(vl-string->list string)

Function-Handling

(apply function list)
(defun sym ([args] [variables]) expr ...)
(defun-q sym ([args] [variables]) expr ...)

(defun-q-list-ref 'function)
(defun-q-list-set 'sym list)
(eval expr)
(lambda arguments expr ...)
(progn [expr] ...)
(trace function ...)
(untrace function ...)

Application-Handling

(arx)
(arxload application [onfailure])
(arxunload application [onfailure])
(autoarxload filename cmdlist)
(autoload filename cmdlist)
(initdia [dialogflag])
(load filename [onfailure])
(startapp appcmd_file)
(vl-load-all filename)
(vl-vbload filename)
(vl-vbarun macroname)
(vlax-add-cmd global-name 'func-sym [local-name cmd-flags])^{COM}

Utility Functions

Query & Command

(acad_colorldg colornum [flag])
(acad_helpdlg helpfile topic)
(command [arguments] ...)
(getcfg cfname)
(getcname cname)
(getenv "variable-name")
(getvar varname)
(help [helpfile [topic [command]]])
(setcfg cfname cfgoal)
(setenv "varname" "value")
(setfunhelp "c:fname" ["helpfile" ["topic" ["command"]]])
(setvar varname value)
(ver)
(vl-cmdf [arguments] ...)
(vlax-add-cmd global-name 'func-sym [local-name cmd-flags])^{COM}
(vlax-remove-cmd global-name)^{COM}

Display Control

(graphscr)
(grdraw from to color [highlight])
(grtext [box text [highlight]])
(grvecs vlist [trans])
(menucmd string)
(menugroup groupname)
(prin1 [expr [file-desc]])
(princ [expr [file-desc]])
(print [expr [file-desc]])
(prompt msg)
(redraw [ename [mode]])
(terpri)
(textpage)
(textscr)
(vports)

User Input

(entsel [msg])
(getangle [pt] [msg])
(getcorner pt [msg])
(getdist [pt] [msg])

(getfiled title default ext flags)
(getint [msg])
(getkword [msg])
(getorient [pt] [msg])
(getpoint [pt] [msg])
(getreal [msg])
(getstring [cr] [msg])
(initget [bits] [string])
(nentsel [msg])
(nentselp [pt])

Geometric

(angle pt1 pt2)
(distance pt1 pt2)
(inters pt1 pt2 pt3 pt4 [onseg])
(osnap pt mode)
(polar pt angle dist)
(textbox elist)

Conversion

(angtof string [mode])
(angtos angle [mode [precision]])
(ascii string)
(atof string)
(atoi string)
(chr integer)
(cvunit value from to)
(distof string [mode])
(itoa int)
(rtos number [mode [precision]])
(trans pt from to [disp])

Device Access

(gread [track] [allkeys [curtype]])
(tablet code [row1 row2 row3 direction])

File-Handling

(close file-desc)
(findfile filename)
(open filename mode)
(read-char [file-desc])
(read-line [file-desc])
(vl-directory-files [directory pattern directories])
(vl-file-copy "source-filename" "dest-filename" [append?])
(vl-file-delete "filename")
(vl-file-directory-p "filename")
(vl-file-rename "old-filename" "new-filename")
(vl-file-size "filename")
(vl-file-systime "filename")
(vl-filename-base "filename")
(vl-filename-directory "filename")
(vl-filename-extension "filename")
(vl-filename-mktemp ["pattern" "directory" "extension"])
(write-char num [file-desc])
(write-line string [file-desc])

Selection Set, Object & Symbol Table Functions

Selection Set Manipulation

(ssadd [ename [ss]])
(ssdel ename ss)

(ssget [mode] [pt1 [pt2]] [pt-list] [filter-list])
(ssgetfirst)
(sslength ss)
(ssmemb ename ss)
(ssname ss index)
(ssname ss index)
(sssetfirst gripset [pickset])

Object (Entity)-Handling

(entdel ename)
(entget ename [applist])
(entlast)
(entmake [elist])
(entmakex [elist])
(entmod elist)
(entnext [ename])
(entupd ename)
(handent handle)
(vlax-dump-object obj [T])^{COM}
(vlax-erased-p obj)^{COM}
(vlax-get-acad-object)^{COM}
(vlax-method-applicable-p obj method)^{COM}
(vlax-object-released-p obj)^{COM}
(vlax-read-enabled-p obj)^{COM}
(vlax-release-object obj)^{COM}
(vlax-typeinfo-available-p obj)^{COM}
(vlax-write-enabled-p obj)^{COM}

Extended Data-Handling

(regapp application)
(xdroom ename)
(xdsite lst)

Symbol Table & Dictionary-Handling

(dictadd ename symbol newobj)
(dictnext ename symbol [rewind])
(dictremove ename symbol)
(dictrename ename oldsym newsym)
(dictsearch ename oldsym newsym)
(layoutlist)
(namedobjdict)
(setview view-desc [viewport-id])
(snvalid symname)
(tblnext table-name [rewind])
(tblobjname table-name symbol)
(tblsearch table-name symbol [setnext])
(vlax-ldata-get dict key [default-data])^{COM}
(vlax-ldata-list dict)^{COM}
(vlax-ldata-put dict key data)^{COM}
(vlax-ldata-test data)^{COM}

Memory Management Functions

(alloc int)
(expand number)
(gc)
(mem)

Reactor Functions^{COM}

(vl-load-com)
(vlr-acdb-reactor data callbacks)
(vlr-add obj)
(vlr-added-p obj)

(vlr-beep-reaction [args])
(vlr-command-reactor data callbacks)^{A02}
(vlr-current-reaction-name)
(vlr-data obj)
(vlr-data-set obj data)
(vlr-deeclone-reactor data callbacks)^{A02}
(vlr-docmanager-reactor data callbacks)^{A02}

(vlr-dwg-reactor data callbacks)^{A02}
(vlr-dxf-reactor data callbacks)^{A02}
(vlr-editor-reactor data callbacks)
(vlr-insert-reactor data callbacks)^{A02}
(vlr-linker-reactor data callbacks)
(vlr-lisp-reactor data callbacks)^{A02}
(vlr-miscellaneous-reactor data callbacks)^{A02}
(vlr-mouse-reactor data callbacks)^{A02}
(vlr-notification reactor owners data callbacks)
(vlr-object-reactor owners data callbacks)
(vlr-owner-add reactor owner)
(vlr-owner-remove reactor owner)
(vlr-owners reactor)
(vlr-pers reactor)
(vlr-pers-p reactor)
(vlr-pers-release reactor)
(vlr-reaction-names reactor-type)
(vlr-reaction-set reactor event function)
(vlr-reactions reactor)
(vlr-reactors reactor-type)
(vlr-remove reactor)
(vlr-remove-all reactor-type)
(vlr-set-notification reactor range)^{A02}
(vlr-sysvar-reactor data callbacks)^{A02}
(vlr-toolbar-reactor data callbacks)^{A02}
(vlr-trace-reaction arguments)
(vlr-type reactor)
(vlr-types)
(vlr-undo-reactor data callbacks)^{A02}
(vlr-wblock-reactor data callbacks)^{A02}
(vlr-window-reactor data callbacks)^{A02}
(vlr-xref-reactor data callbacks)^{A02}

VLX Namespace Functions

(vl-arx-import [function | application])
(vl-doc-export 'function)
(vl-doc-import ['func | application])
(vl-doc-ref symbol)
(vl-doc-set symbol value)
(vl-exit-with-error "msg")
(vl-exit-with-value value)
(vl-list-exported-functions ["appname"])
(vl-list-loaded-vlx)
(vl-unload-vlx)
(vl-vlx-loaded-p "appname")

Namespace Communication Functions

(vl-bb-ref variable)
(vl-bb-set variable value)
(vl-load-all "filename")
(vl-propagate variable)

Windows Registry Functions

(vl-registry-delete reg-key [val-name])

(vl-registry-descendents reg-key [val-names])
(vl-registry-read reg-key [val-name])
(vl-registry-write reg-key [val-name val-data])
(vlax-product-key)^{COM}

Visual LISP Extensions to AutoLISP^{COM}

Collection Manipulation

(vlax-for sym collection [expr1 [expr2 ...]])
(vlax-map-collection obj function)

Data Conversion

(vlax-3D-point list)
(vlax-3D-point x y [z])
(vlax-ename->vla-object entname)
(vlax-make-safearray type [(l-bound . u-bound) [(l-bound . u-bound) ...]])^{A02}
(vlax-make-variant value type)
(vlax-safearray-fill var 'element-value)
(vlax-safearray-get-dim var)
(vlax-safearray-get-element var element)
(vlax-safearray-get-l-bound var dim)
(vlax-safearray-get-u-bound var dim)
(vlax-safearray-put-element var element value)

(vlax-safearray-type var)
(vlax-safearray->list var)
(vlax-tmatrix list)
(vlax-variant-change-type var type)
(vlax-variant-type var)
(vlax-variant-value var)
(vlax-vla-object->ename obj)

Method & Property-Handling

(vlax-invoke-method obj method list)
(vlax-method-applicable-p obj method)
(vlax-get-property obj property)
(vlax-property-available-p obj prop [T])
(vlax-put-property obj property arg)
(vla-method object arg1 arg2 ...)
(vla-get-property object)
(vla-put-property object new-value)

Object-Handling

(vlax-create-object "prog-id")
(vlax-dump-object obj [T])
(vlax-erased-p obj)
(vlax-get-acad-object)
(vlax-get-object "prog-id")
(vlax-get-or-create-object "prog-id")
(vlax-import-type-library :tbl-filename filename
:methods-prefix mprefix
:properties-prefix pprefix
:constants-prefix cprefix)
(vlax-method-applicable-p obj method)
(vlax-object-released-p obj)
(vlax-read-enabled-p obj)
(vlax-release-object obj)
(vlax-typeinfo-available-p obj)
(vlax-write-enabled-p obj)

Curve Measurement

(vlax-curve-getArea curve-obj)

(vlax-curve-getDistAtParam curve-obj param)
(vlax-curve-getDistAtPoint curve-obj pt)
(vlax-curve-getEndParam curve-obj)
(vlax-curve-getEndPoint curve-obj)
(vlax-curve-getParamAtDist curve-obj param)
(vlax-curve-getParamAtPoint curve-obj pt)
(vlax-curve-getPointAtDist curve-obj dist)
(vlax-curve-getPointAtParam curve-obj param)
(vlax-curve-getStartParam curve-obj)
(vlax-curve-getStartPoint curve-obj)
(vlax-curve-isClosed curve-obj)
(vlax-curve-isPeriodic curve-obj)
(vlax-curve-isPlanar curve-obj)
(vlax-curve-getClosestPointTo curve-obj givenPt [extend])
(vlax-curve-getClosestPointToProjection curve-obj givenPt normal [extend])
(vlax-curve-getFirstDeriv curve-obj param)
(vlax-curve-getSecondDeriv curve-obj param)

Dictionary

(vlax-ldata-delete dict key)
(vlax-ldata-get dict key [default-data])
(vlax-ldata-list dict)
(vlax-ldata-put dict key data)
(vlax-ldata-test data)

Notes:

- To refresh (the display of) an object after updating its property, issue this command: (vla-update object)
- All ActiveX and VBA predefined constants can be used in VLISP expressions. For example: (vla-put-color mycircle acRed). See *ActiveX and VBA Reference* and *Connectivity Automation Reference*.

LEGENDS

Item^{A02} New in ACAD 2002
Item^{A04} New in ACAD 2004
Item^{COM} Requires (vl-load-com)

Reactor Events and Callback Data

Reactor Types

:VLR-AcDb-Reactor
 :VLR-DocManager-Reactor
 :VLR-Command-Reactor
 :VLR-DeepClone-Reactor
 :VLR-DWG-Reactor
 :VLR-DXF-Reactor
 :VLR-Insert-Reactor
 :VLR-Miscellaneous-Reactor
 :VLR-Mouse-Reactor
 :VLR-SysVar-Reactor
 :VLR-Toolbar-Reactor
 :VLR-Undo-Reactor
 :VLR-Wblock-Reactor
 :VLR-Widow-Reactor
 :VLR-XREF-Reactor
 :VLR-Linker-Reactor
 :VLR-Object-Reactor

AcDb (Database) reactor events

:vlr-objectAppended
 :vlr-objectUnAppended
 :vlr-objectReAppended
 :vlr-objectOpenedForModify
 :vlr-objectModified
 :vlr-objectErased
 :vlr-objectUnErased

Command reactor events

:vlr-unknownCommand
 :vlr-commandWillStart
 :vlr-commandEnded
 :vlr-commandCancelled
 :vlr-commandFailed

DeepClone reactor events

:vlr-beginDeepClone
 :vlr-beginDeepCloneXlation¹
 :vlr-abortDeepClone
 :vlr-endDeepClone

DocManager reactor events

:vlr-documentCreated¹
 :vlr-documentToBeDestroyed¹
 :vlr-documentLockModeWillChange³
 :vlr-documentLockModeChangeVetoed²
 :vlr-documentLockModeChanged³
 :vlr-documentBecameCurrent¹
 :vlr-documentToBeActivated¹
 :vlr-documentToBeDeactivated¹

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:
¹ **integer**: error status.
 :vlr-documentCreated¹
 :vlr-documentToBeDestroyed¹
 :vlr-documentLockModeWillChange³
 :vlr-documentLockModeChangeVetoed²
 :vlr-documentLockModeChanged³
 :vlr-documentBecameCurrent¹
 :vlr-documentToBeActivated¹
 :vlr-documentToBeDeactivated¹
 ☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:
¹ **VLA-obj**: the affected doc obj;
² **VLA-obj**: the affected doc obj;
string: global cmd string passed in. Prefixed with "#" if the callback is being made on behalf of an unlock request.
³ **VLA-obj**: the affected doc obj;
integer: lock in effect;
integer: lock mode after the lock is applied;
integer: strongest lock mode from all other exec contexts;
string: global cmd string passed in. Prefixed with "#" if it is an unlock request.
 Lock modes are:
 1- Auto Write Lock
 2- Not Locked

4- Shared Write
 8- Read
 10- Exclusive Write

DWG reactor events

:vlr-beginClose
 :vlr-databaseConstructed
 :vlr-databaseToBeDestroyed
 :vlr-beginDwgOpen¹
 :vlr-endDwgOpen¹
 :vlr-dwgFileOpened¹
 :vlr-beginSave²
 :vlr-saveComplete³

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:
¹ **string**: file to open
² **string**: default file name for save.
³ **string**: actual file name used for save.

DXF reactor events

:vlr-beginDxfIn
 :vlr-abortDxfIn
 :vlr-dxfInComplete
 :vlr-beginDxfOut
 :vlr-abortDxfOut
 :vlr-dxfOutComplete

☛ Callback args: reactor obj, list of extra data.

Insert reactor events

:vlr-beginInsert¹
 :vlr-beginInsertM²
 :vlr-otherInsert³
 :vlr-endInsert⁴
 :vlr-abortInsert⁴

☛ Callback args: reactor obj, list of extra data.

☛ Callback data:
¹ **VLA-obj**: the db in which the block is being inserted;
string: the block to be inserted;
VLA-obj: the source db of the block.
² **VLA-obj**: the target db;
matrix: the 3D transformation matrix;
VLA-obj: the source db of the matrix.
³ **VLA-obj**: the target db;
VLA-obj: the source db of the block or matrix.
⁴ **VLA-obj**: the target db.

Linker reactor events

:vlr-rxAppLoaded
 :vlr-rxAppUnLoaded

☛ Callback args: reactor obj, list of ObjectARX program names.

LISP reactor events

:vlr-lispWillStart¹
 :vlr-lispEnded
 :vlr-lispCancelled

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:
¹ **string**: the 1st line of the AutoLISP exp to eval.

Miscellaneous reactor events

:vlr-pickFirstModified
 :vlr-layoutSwitched¹

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:
¹ **string**: the layout switched to.

Mouse reactor events

:vlr-beginDoubleClick¹
 :vlr-beginRightClick¹

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:
¹ **3D pt list**: the point clicked on, in WCS.

Object events

:vlr-cancelled
 :vlr-copied¹
 :vlr-erased
 :vlr-unerased

:vlr-goodbye
 :vlr-openedForModify
 :vlr-modified
 :vlr-subObjModified²
 :vlr-modifiedUndone
 :vlr-modifiedXData
 :vlr-unappended
 :vlr-reappended
 :vlr-objectClosed

☛ Callback args: owner, reactor obj, list of extra data.
 ☛ Callback data:
¹ **ename**: the object created by the copy operation.
² **ename**: the sub-object that has been modified.

SysVar reactor events

:vlr-sysVarWillChange¹
 :vlr-sysVarChanged²

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:
¹ **string**: the sysvar name.
² **string**: the sysvar name;
Tnil: whether the changed was successful.

Toolbar reactor events

:vlr-toolbarBitmapSizeWillChange¹
 :vlr-toolbarBitmapSizeChanged¹

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:
¹ **Tnil**: whether the toolbar is set to large bitmaps.

Undo reactor events

:vlr-undoSubcommandAuto¹
 :vlr-undoSubcommandControl²
 :vlr-undoSubcommandBegin³
 :vlr-undoSubcommandEnd³
 :vlr-undoSubcommandMark³
 :vlr-undoSubcommandBack³
 :vlr-undoSubcommandNumber⁴

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:
¹ **integer**: the activity (always 4);
symbol: the state of Auto mode. T if turned on.
² **integer**: the activity (always 4);
integer: selected Control option. Possible values:
 0- NONE was selected
 1- ONE was selected
 2- ALL was selected
³ **integer**: always 0.
⁴ **integer**: the activity (always 0);
integer: the number of steps being undone.

Wblock reactor events

:vlr-wblockNotice¹
 :vlr-beginWblockPt²
 :vlr-beginWblockId³
 :vlr-beginWblock⁴
 :vlr-otherWblock⁴
 :vlr-abortWblock⁵
 :vlr-endWblock⁵

:vlr-beginWblockObjects⁶

☛ Callback args: reactor obj, list of extra data.

☛ Callback data:
¹ **VLA-obj**: db object from which the block will be created.
² **VLA-obj**: the target db;
VLA-obj: the source db;
3D pt list: the base point in the target db, in WCS.
³ **VLA-obj**: the target db;
VLA-obj: the source db;
object ID of the block TableRecord being wblocked.
⁴ **VLA-obj**: the target database;
VLA-obj: the source database.
⁵ **VLA-obj**: the target database.
⁶ **VLA-obj**: the source database;
ID map.

Window reactor events

:vlr-docFrameMovedOrResized¹
 :vlr-mainFrameMovedOrResized¹

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:

¹ **integer**: the HWnd of the window;
Tnil: whether the window has been moved or resized.

XREF reactor events

:vlr-beginAttach¹
 :vlr-otherAttach²
 :vlr-abortAttach³
 :vlr-endAttach⁴
 :vlr-redirected⁵
 :vlr-comandeered⁶
 :vlr-beginRestore⁷
 :vlr-abortRestore⁸
 :vlr-endRestore⁸
 :vlr-xrefSubcommandBindItem⁹
 :vlr-xrefSubcommandAttachItem¹⁰
 :vlr-xrefSubcommandOverlayItem¹¹
 :vlr-xrefSubcommandDetachItem¹²
 :vlr-xrefSubcommandPathItem¹³
 :vlr-xrefSubcommandReloadItem¹⁴
 :vlr-xrefSubcommandUnloadItem¹⁵

☛ Callback args: reactor obj, list of extra data.
 ☛ Callback data:

¹ **VLA-obj**: target drawing db;
string: filename of the xref;
VLA-obj: source drawing db.
² **VLA-obj**: target drawing db;
VLA-obj: source drawing db.
³ **VLA-obj**: source drawing db (contains the objs being attached).
⁴ **VLA-obj**: target drawing db.
⁵ **integer**: obj ID for the redirected symbol table record (STR) in the drawing XREFed to;
integer: object ID for the object in the xref drawing.
⁶ **VLA-obj**: target db;
integer: object ID;
VLA-obj: source drawing db.
⁷ **VLA-obj**: target drawing db;
string: XREF block tbl record (BTR) name;
VLA-obj: source drawing db.
⁸ **VLA-obj**: target drawing db.
⁹ **integer**: activity the BIND is carrying out. Values:
 0- BIND subcmd invoked.
 2- xref with indicated obj ID is begin bound.
 3- xref with indicated obj ID was successfully bound.
 4- BIND subcmd completed.
 5- BIND operation is about to terminated or fail.
 6- BIND operation has terminated or failed.
 7- Sent for XDep block bound by XBind.
 8- Sent for all other symbols: Layers, Linetypes, TextStyles, and DimStyles.
integer: obj ID for the xref being bound, or 0 if not applicable.
¹⁰ **integer**: activity the ATTACH is carrying out. Possible values are 0, 2-6, see above.
string: file being attached, or nil if not applicable.
¹¹ **integer**: activity the OVERLAY is carrying out. Possible values are 0, 2-6, see above.
string: file being overlaid, or nil if not applicable.
¹² **integer**: activity the DETACH is carrying out. Possible values are 0, 2-6, see above.
string: file being detached, or nil if not applicable.
¹³ **integer**: activity the OVERLAY is carrying out. Possible values are 0, 2-6, see above.
integer: obj ID of the xref being operated on, or 0 if not applicable;
string: new path name of the xref, or nil if not applicable.
¹⁴ **integer**: activity the RELOAD is carrying out. Possible values are 0, 2-6, see above.
integer: obj ID of the xref being reloaded, or 0 if not applicable.
¹⁵ **integer**: activity the UNLOAD is carrying out. Possible values are 0, 2-6, see above.
string: obj ID of the xref being unloaded, or 0 if not applicable.

Externally Defined Commands

(c:3dsin *mode* [*multimat create*] *file*)
 (c:3dsout *sset omode div smoth weld file*)
 (align *arg1 arg2*)
 (c:cal *expression*)
 (c:fog *enabled* [*color* [*near-dist*] [*far-dist*] [*near-percent*] [*far-percent*]]])
 (c:light *mode* [*options*])
 (c:lscedit *mode* [*options*])

(c:slib *mode* [*options*])
 (c:lsnew *obj-type height position alignment*)
 (c:matlib *mode name* [*file*])
 (mirror3d *arg1 arg2* ...)
 (c:render [*filename* | *pt1 pt2*])
 (c:renderupdate [*RU-value*])
 (c:replay *filename type* [*xoff*] *yoff* *xsize* *ysize*)
 (c:rmat *mode options*)
 (rotate3d *args* ...)
 (c:rpref *mode option* [*setting*])
 (c:saveimg *filename type* [*portion*] [*xoff*] *yoff* *xsize* *ysize*] [*compression*])
 (c:scene *mode* [*options*])
 (c:setuv *mode options*)
 (c:showmat *arg*)
 (c:solprof *args*)
 (c:stats [*filename* | *nil*])

VLAX Variant Types & Assoc. LISP Datatype

0	vlax-vbEmpty	nil
1	vlax-vbNull	:vlax-null
2	vlax-vbInteger	integer
3	vlax-vbLong	
4	vlax-vbSingle	
5	vlax-vbDouble	real
8	vlax-vbString	string
9	vlax-vbObject	VLA-object
11	vlax-vbBoolean	:vlax-true/false
8192+n	vlax-vbArray	vlax safearray

System Variables

ACADVER: ACAD version
 ANGBASE: base angle orientation
 AUNITS, AUPREC: angle unit style and precision. Values are:
 0- Degrees
 1- Degrees/Minutes/Seconds
 2- Grads
 3- Radians
 4- Surveyor's units
 CDATE: system date/time in readable format
 CLAYER: current layer name
 CMDACTIVE: whether a cmd is active?
 CMDECHO: whether cmd echo is on?
 CMDNAMES: cmd names currently active
 CTAB: current (model/space layout) tab
 DATE: system date/time in comp format
 DWGNAME: drawing name
 DWGPREFIX: folder where drawing is located
 EXPERT: expert mode
 LUNITS, LUPREC: non-angle number unit style and precision. Values are:
 1- Scientific notation
 2- Decimal
 3- Engineering (ft & dec. in.)
 4- Architectural (ft. & frac. in.)
 5- Fractions
 OSMODE: object snap modes (bit-encoded integer)

_PKSER: package serial number
 TDCREATE: date/time, in Julian day format, when the drawing is created
 TDINDWG: time in days that the drawing has been worked on
 TEXTSTYLE: current text style name
 UCSNAME: UCS name
 UNITMODE: 0 (default): loose; 1: compact.

Extended Data (XData)

Group Codes

1000 String. A string up to 255 bytes.
 1001 **Application name**. A string up to 31 bytes. It is the beginning of a new application extended data group.
 1002 **Control string**. A string, either "{" or "}".
 1003 **Layer name**. A string.
 1004 Binary data, up to 127 bytes. In ASCII DXF files, binary data is represented as a string of hexadecimal digits, two per binary byte.
 1005 **Database handle**. Refers to another entity.
 1010, 1020, 1030 Three real values, in the order X, Y, Z. Never altered.
 1011, 1021, 1031 World space position. A 3D point to be altered.
 1012, 1022, 1032 World space displacement. A 3D point to be altered.
 1013, 1023, 1033 World direction. A 3D point to be altered.
 1040 Real. A real value.
 1041 Distance. A real value that is scaled along with the parent entity.
 1042 Scale factor. A real value that is scale along with the parent.
 1070 Integer. A 16-bit integer (signed or unsigned).
 1071 Long. A 32-bit integer.

Example:
 (-3
 ("AppName1" (1000 . "HATCH") (1002 . "{")
 (1070 . 16) (1000 . "LINE") (1040 . 1.0)
 (1040 . 0.0) (1002 . "}")
 ("AppName2" (...))
); end of xdata

XRecord Group Codes

100 Subclass marker (AcDbXrecord)
 1-369 (except 5 and 105) Can be used in any way.