



Bricscad V12.2.5 VLE LISP Function Summary

Function	Type	Form	Purpose	Arguments	Returns	Example	Notes
VLE-ENAMEP	DataType	(vle-enamep obj)	Predicator function to determine whether 'obj' is of ENAME type	'obj' any Lisp item to be verified	T - if 'obj' is a ENAME object NIL - if 'obj' is not a ENAME object		Replacement for : (= (type obj) 'ENAME) and similar
VLE-FILEP	DataType	(vle-filep obj)	Predicator function to determine whether 'obj' is of FILE type	obj' any Lisp item to be verified	T - if 'obj' is a FILE object NIL - if 'obj' is not a FILE object		Replacement for : (= (type obj) 'FILE) and similar
VLE-INTEGERP	DataType	(vle-integerp obj)	Predicator function to determine whether 'obj' is of integer type	'obj' any Lisp item to be verified	T - if 'obj' is a integer object NIL - if 'obj' is not a integer object		Replacement for : (= (type obj) 'INT) and similar
VLE-NUMBERP	DataType	(vle-numberp obj)	Predicator function to determine whether 'obj' is of any number type (int double)	'obj' any Lisp item to be verified	T - if 'obj' is a number object NIL - if 'obj' is not a number object		Identical to (numberp), implemented for completeness
VLE-PICKSETP	DataType	(vle-picksetp obj)	Predicator function to determine whether 'obj' is of PICKSET type	obj' any Lisp item to be verified	T - if 'obj' is a PICKSET object NIL - if 'obj' is not a PICKSET object		Replacement for : (= (type obj) 'PICKSET) and similar
VLE-REALP	DataType	(vle-realp obj)	Predicator function to determine whether 'obj' is of REAL (double) type	'obj' any Lisp item to be verified	T - if 'obj' is a REAL (double) object NIL - if 'obj' is not a REAL (double) object		Replacement for : (= (type obj) 'REAL) and similar
VLE-STRINGP	DataType	(vle-stringp obj)	Predicator function to determine whether 'obj' is of STRING type	'obj' any Lisp item to be verified	T - if 'obj' is a string object NIL - if 'obj' is not a string object		Replacement for : (= (type obj) 'STR) and similar
VLE-VARIANTP	DataType	(vle-variantp obj)	Predicator function to determine whether 'obj' is of VARIANT type	'obj' any Lisp item to be verified	T - if 'obj' is a VARIANT object NIL - if 'obj' is not a VARIANT object		Replacement for : (= (type obj) 'VARIANT) and similar
VLE-VLAOBJECTP	DataType	(vle-vlaobjectp obj)	Predicator function to determine whether 'obj' is of VLA-OBJECT type	'obj' any Lisp item to be verified	T - if 'obj' is a VLA-OBJECT object NIL - if 'obj' is not a VLA-OBJECT object		Replacement for : (= (type obj) 'VLA-OBJECT) and similar
VLE-ENTGET	Entity	(vle-entget dxf ename)	Retrieves the entity's property value for specified DXF group code in typical case, this is a high-performance replacement for usual code like (cdr (assoc DXF (entget ename)))	'dxf' the DXF group code to retrieve  'ename' the entity's ename	the value for specified DXF group code as plain Lisp data	(vle-entget 62 entity) returns the color for entity	
VLE-ENTMOD	Entity	(vle-entmod dxf ename value)	Modifies the entity's property value for specified DXF group code in typical case, this is a high-performance replacement for usual code like (entmod (subst (cons dxf value) olditem lst))	'dxf' the DXF group code to set the new value for  'ename' the entity's ename  'value' new value to be set for the entity's property (can be plain Lisp data or VLA/VLAX data value)	T if the entity' value has been updated (or is identical) NIL if updating the entity's DXF value failed	(vle-entmod 62 entity 112) sets the color index as 112	
VLE-FILE->LIST	File	(vle-file->list filename commentchar)	Loads the text lines of specified file as a list of strings	'filename' name of the file to be loaded  'commentchar' the "comment" character to ignore comment lines	List of file lines; or NIL, if the file is not found or empty		'filename' is searched in all support paths, if it does not contain a path; 'commentchar' only used if not NIL; if NIL, all file lines are loaded



Bricscad V12.2.5 VLE LISP Function Summary

Function	Type	Form	Purpose	Arguments	Returns	Example	Notes
VLE-APPEND	List	(vle-append lst item)	Returns a list with given item added to it.	'lst' normal Lisp list  'item' item to append to 'lst'	List of items, with new item appended	(vle-append '("bread" "cheese" "milk") "jam") returns ("bread" "cheese" "milk" "jam")	replacement for (append '("bread" "cheese" "milk") '("jam"))).
VLE-CASSOC	List	(vle-cassoc key lst)	See VLE-CDRASSOC				
VLE-CDRASSOC	List	(vle-cdrassoc key lst)	searches 'lst' list for a dotted-pair with key 'key' and returns its value; replacement for (cdr (assoc key lst))	'key' key value for the dotted pair value to search for  'lst' list of dotted pairs	The value of dotted pair (key . value), if a dotted pair with key 'key', if present in the list; NIL if not present	(vle-cdrassoc 11 '((1 . "a")(2 . "b"))(11 . "xx"))(22 . "yy")) returns "xx"	Alias: vle-get-cdrassoc vle-get-cassoc vle-cassoc
VLE-FIND	List	(vle-find item lst)	See VLE-MEMBER				
VLE-GET-CASSOC	List	(vle-get-cassoc key lst)	See VLE-CDRASSOC				
VLE-GET-CDRASSOC	List	(vle-get-cdrassoc key lst)	See VLE-CDRASSOC				
VLE-MEMBER	List	(vle-member item lst)	verifies whether list 'lst' contains the expression 'item' replacement for (member item lst)	'item' expression to search for in 'lst'  'lst' normal Lisp list	T if the expression 'item' is found in list 'lst'; NIL if not found	Alias: vle-find	
VLE-NTH0	List	(vle-nth<x> lst)	returns the item at index <x> of the list 'lst'; relacement for (nth <x> lst), where <x> is an interger from 0 - 9.	'lst' list of items	item at index <x> of the input list; index is 0-based !	(vle-nth4 '(0 11 22 33 44 55 66 77 88 99) returns 44 (vle-nth8 '(0 11 22 33 44 55 66 77 88 99) returns 88	
VLE-NTH1	List	(vle-nth<x> lst)	See VLE-NTH0				
VLE-NTH2	List	(vle-nth<x> lst)	See VLE-NTH0				
VLE-NTH3	List	(vle-nth<x> lst)	See VLE-NTH0				
VLE-NTH4	List	(vle-nth<x> lst)	See VLE-NTH0				
VLE-NTH5	List	(vle-nth<x> lst)	See VLE-NTH0				
VLE-NTH6	List	(vle-nth<x> lst)	See VLE-NTH0				
VLE-NTH7	List	(vle-nth<x> lst)	See VLE-NTH0				
VLE-NTH8	List	(vle-nth<x> lst)	See VLE-NTH0				
VLE-NTH9	List	(vle-nth<x> lst)	See VLE-NTH0				
VLE-REMOVE	List	(vle-remove item lst)	See VLE-REMOVE-FIRST				Removed. Was available in Beta, replaced by VLE-REMOVE-FIRST, which better reflects the purpose of this function.
VLE-REMOVE-ALL	List	(vle-remove-all item lst)	Removes all occurences of 'item' from list 'lst'	'item' object to remove from 'lst'  'lst' normal Lisp list	list 'lst' without all occurences of 'item', if found; unchanged list 'lst' if not found	(vle-remove-all 123 '(1 2 123 4 5 123 6 7 8)) returns '(1 2 4 5 6 7 8)	
VLE-REMOVE-FIRST	List	(vle-remove-first item lst)	Removes the first occurence of 'item' from list 'lst'	'item' object to remove from 'lst'  'lst' normal Lisp list	list 'lst' without first occurence of 'item', if found; unchanged list 'lst' if not found	(vle-remove-first 123 '(1 2 123 4 5 123 6 7 8)) returns '(1 2 4 5 123 6 7 8)	
VLE-REMOVE-IDX	List	(vle-remove-idx idx lst)	See VLE-REMOVE-NTH				
VLE-REMOVE-LAST	List	(vle-remove-last lst)	Removes the last item from list 'lst'; replacement for (reverse (cdr (reverse lst)))	'lst' normal Lisp list	list 'lst' without last item		



Bricscad V12.2.5 VLE LISP Function Summary

Function	Type	Form	Purpose	Arguments	Returns	Example	Notes
VLE-REMOVE-NTH	List	(vle-remove-nth idx lst)	Removes the object at index 'idx' from list 'lst' (index is 0-based)	'idx' index of the item to be removed; if negative, or 'lst' contains less items, nothing is removed  'lst' normal Lisp list	list 'lst' without item at position 'idx'; if 'idx' is not in the list, the unchanged list 'lst' is returned	(vle-remove-nth 3 '(1 2 123 4 5 123 6 7 8)) returns '(1 2 123 5 123 6 7 8)	Alias: vle-remove-idx
VLE-REPLACE	List	(vle-replace lst idx val)	See VLE-SUBST-NTH				
VLE-SAFEARRAYP	List	(vle-safearrayp obj)	Predicator function to determine whether 'obj' is of SAFEARRAY type	'obj' any Lisp item to be verified	T - if 'obj' is a SAFEARRAY object NIL - if 'obj' is not a SAFEARRAY object		Replacement for : (= (type obj) 'SAFEARRAY) and similar
VLE-SEARCH	List	(vle-search item lst asldx)	Searches for 'item' in list 'lst' and returns either the index of 'item' in 'lst', if 'asldx' is non-NIL, or the list starting with 'item' if 'asldx' is NIL, or NIL if 'item' is not contained in 'lst'	'item' object to search for in 'lst'  'lst' list of items to search in  'asldx' flag defining the type of result : if NIL, the list starting with 'item' is returned if non-NIL, the index of 'item' in 'lst' is returned	The index of 'item' in 'lst', or the sub-list starting with 'item', depending on 'asldx' flag; index is always 0-based; if 'item' is not contained in 'lst', NIL is returned	(vle-search 11 '(1 2 3 11 22 33 ) nil) => '(11 22 33) (vle-search 11 '(1 2 3 11 22 33 ) t) => 3	
VLE-SET-CASSOC	List	(vle-set-cassoc key lst val)	See VLE-SET-CDRASSOC				
VLE-SET-CDRASSOC	List	(vle-set-cdrassoc key lst val)	Searches 'lst' list for dotted-pairs with key 'key' and sets the value of the dotted pairs (if found) to 'val'; replacement for (subst (cons key val) (assoc key lst) lst)	'key' key value for the dotted pair value to search for  'lst' list of dotted pairs  'val' new value for dotted pairs with key 'key'	the input list with changed values for dotted pairs with 'key'; if there is no dotted-pair with 'key', 'lst' returns unchanged	(setq lst '((1 . "a")(2 . "b")(1 . "a")(2 . "b"))) (setq lst (vle-set-cdrassoc 1 lst "A")) returns '((1 . "A")(2 . "b")(1 . "A")(2 . "b"))	Alias: vle-set-cassoc
VLE-SUBLIST	List	(vle-sublist lst startidx nritems)	Creates a sublist from given input list 'lst', starting with item at index 'startindex' and of maximum 'nritems' number of list items from 'lst'	'lst' list to extract a sublist from  'startidx' index of first item for sublist (0-based); if 'startindex' is < 0, 0 is used  'nritems' number of items to be copied from 'lst' list if 'nritems' is < 0, all items after 'startindex' are copied to result list	A sublist of 'lst' starting with index 'startidx' and containing 'nritems' of input list 'lst'; if 'lst' has not enough items, only the items of 'lst' are copied to result list, no NIL is appended	(vle-sublist '(1 2 3 4 5 6 7) 2 3) => '(3 4 5) (vle-sublist '(1 2 3 4 5 6 7) 2 -1) => '(3 4 5 6 7) (vle-sublist '(1 2 3 4 5 6 7) 2 10) => '(3 4 5 6 7)	
VLE-SUBST-NTH	List	(vle-subst-nth lst idx val)	Sets / replaces the item at index 'idx' in list 'lst' with specified new value 'val'	'lst' input list where the item at position 'idx' is to be set  'idx' the 0-based index of the item to be replaced by new 'val'  'val' the new value to be set for the item	list 'lst' without item at position 'idx'; if 'idx' is not if there is no item 'idx' in 'lst' the list remains unchanged; if 'idx' is < 0 or greater than the list length, the unchanged list is returned	(vle-subst-nth '(1 2 3 4 5) 2 99) => '(1 2 99 4 5) (vle-subst-nth '(1 2 3 4 5) 10 99) => '(1 2 3 4 5)	Alias: vle-replace
VLE-STRING-SPLIT	String	(vle-string-split keys string)	Modifies the entity's property value for specified DXF group code; in typical case, this is a high-performance replacement for usual code like (entmod (subst (cons dxf value) olditem lst))	'keys' string containing all delimiter characters to be used  'string' the string to be splitted into tokens	list of tokens; if none of the <keys> characters is present in string, the list contains only the original string (1 token)	(vle-string-split " , " "Sample, using different keys; voila") will return ("Sample" " using different keys" " voila")	Alias: vl-string-split, string-split



Bricscad V12.2.5 VLE LISP Function Summary

Function	Type	Form	Purpose	Arguments	Returns	Example	Notes
VLE-EXTENSIONS-ACTIVE	Symbol	N/A	Indicates that VLE extensions are present	N/A	T if VLE is present NIL if VLE is not present	At command line: !VLE-EXTENSIONS-ACTIVE	
VLE-DISPLAYPAUSE	Transaction	(vle-displaypause pauseDisplay)	Modifies the entity's property value for specified DXF group code; in typical case, this is a high-performance replacement for usual code like (entmod (subst (cons dxf value) olditem lst))	pauseDisplay' T disable display updates, NIL enable display updates	T when display updates are disabled NIL when display updates are enabled		This function can be nested - but each call with 't' should be paired by a call with 'nil'; when Lisp execution is finished, the display will be automatically enabled for safety
VLE-END-TRANSACTION	Transaction	(vle-end-transaction)	Finishes the active database transaction for current drawing	None	T when transaction is active, NIL otherwise	(vle-end-transaction) -> t : transaction is active nil : transaction is not active	see (vle-start-transaction) when outermost Lisp execution finishes, a pending / open transaction is automatically closed (i.e. when Lisp code errors)
VLE-START-TRANSACTION	Transaction	(vle-start-transaction)	Starts a database transaction for current drawing	None	T when transaction is active, NIL otherwise	(vle-start-transaction) -> t : transaction is active nil : transaction is not active	With an active transaction, database and display updates are delayed until the transaction is finished - this can significantly improve performance when many entity and database modifications are done; multiple / nested transactions are not supported - but it is gracefully tolerated, Lisp only keeps a single (global) transaction; when outermost Lisp execution finishes, a pending / open transaction is automatically closed (i.e. when Lisp code errors)



Bricscad V12.2.5 VLE LISP Function Summary

Function	Type	Form	Purpose	Arguments	Returns	Example	Notes
VLE-ALERT	Utility	(vle-alert title msg flags)	shows a message box, which can be customised in wide range (based on Windows ::MessageBox() function)	'title' title for the message box (the caption string) 'msg' the message to be displayed; will be word-wrapped 'flags' combination (addition) of integers specifying behaviour Flags for button : 0 MB_OK 1 MB_OKCANCEL 2 MB_ABORTRETRYIGNORE 3 MB_YESNOCANCEL 4 MB_YESNO 5 MB_RETRYCANCEL 6 MB_CANCELTRYCONTINUE Flags for icons : 16 MB_ICONHAND / MB_ICONSTOP 32 MB_ICONQUESTION 48 MB_ICONEXCLAMATION 64 MB_ICONASTERISK / MB_ICONINFORMATION Flags for default button (for <return> : 0 MB_DEFBUTTON1 256 MB_DEFBUTTON2 512 MB_DEFBUTTON3 768 MB_DEFBUTTON4 Flags for behaviour : 0 MB_APPLMODAL 4096 MB_SYSTEMMODAL 8192 MB_TASKMODAL 16384 MB_HELP 65536 MB_SETFOREGROUND 131072 MB_DEFAULT_DESKTOP_ONLY 262144 MB_TOPMOST 524288 MB_RIGHT 1048576 MB_RTLCREADING	Number of button which was used to finish the dialog	(vle-alert "My CAD App" "Dear Customer ...." (+ 4 32 4096)) will show Yes+No button, the question mark icon, as system modal dialog	
VLE-COLLECTION->LIST	Utility	(vle-collection->list collection)	Creates a normal list of items from given COM collection	'collection' the collection to get the items as normal list from	list of items, or NIL if collection is empty		
VLE-SELECTIONSET->LIST	Utility	(vle-selectionset->list selectionset)	Creates a normal list of entities from given selection set	selectionset' the selection set to get as entities list from	list of plain entity names, or NIL if selectionset is empty	(vle-selectionset->list ss) -> (<#Entity name> <#Entity name>)	
VLE-STARTAPP	Utility	(vle-startapp cmd args mode)	Runs the specified command, optionally with arguments, and optionally waits for the process to end (synchronous execution); replaces (startapp), which does not wait for the process (asynchronous)	'cmd' the command / program to run (*.exe)  'args' optional arguments as string, or NIL if no arguments  'mode' T specifies to wait for the process, NIL runs parallel	process ID if started parallel, or the program's return status	(vle-startapp "notepad.exe" "test.dat" t)	