

Install Library

If you click INSTALL LIBRARY, a Library menu is added before the Property menu that contains the following commands:

Enter Guide	Use this to make pressing Enter move the cell selector. Page: entguid
Delete Negatives	Use this to delete negative values from the active block. Page: delneg
Dates to Labels	Use this to convert dates in the active block into labels. Page: datelab
Make Pipes	Use this to search a file for commas and replace them with pipes (). Page: makepipe
Quick Search	Use this to quickly search for a string. A shortcut for this macro is Ctrl+S. You can use Ctrl+N to move between strings found by the search. Page: commeq
Get Phone Number	Use this to display a form for entering a name and phone number. Page: commeq
Get Tasks	Use this to link up with the ObjectVision application TASKLIST.OVD and store the tasks listed there in the active notebook (starting at the current cell). Page: ddeuse
Make Macro Buttons	Use this to create macro buttons. Each cell in the selected block that contains a label has a button created over it. Page: makebuttons
Cell Typer	Use this to display the cell type of the current cell (Label, Value, Blank) on the status line. (Press Esc to stop the macro.) Page: indicell
Remove Library	Use this to remove the Library menu and close this notebook. Page: startup

PgDn

PgUp

PgDn

PgUp

```
_remove_ma{WindowShow "MACROS.WB1"}  
          {DELETEMENU "/Library"}  
          {FileClose 0}
```

col_count	5	rows	9
cell_address	[NOTEBK1]\$A:\$C\$8	columns	4
cell_format	G	row_count	10
date_block	[]A:A1..A:E10		

date_lab	{LET date_block,+"[]"&@PROPERTY("Active_Block.Selection")}	Name the active block date_block
	{LET rows, @ROWS(@@(date_block))-1}	Store the height of the block
	{LET columns, @COLS(@@(date_block))-1}	Store the width of the block
	{FOR row_count,0,rows,1,check_rows}	Check each row in the block for dates
	{QUIT}	Exit the macro

check_rows	{FOR col_count,0,columns,1,check_date}{RETURN}	Check each cell in the row
------------	--	----------------------------

check_date	{LET cell_format,@CELLINDEX("format",@@(date_block),col_count,row_count)}
	{IF @LEFT(cell_format,1)="D">{BRANCH convert_cell}}

convert_cell	{LET cell_address, +@CELLINDEX("address",@@(date_block),col_count,row_count)}
	{CONTENTS +cell_address,+cell_address}
	{RETURN}

store the current cells format as a variable
check if it is a date, call convert_cell if it is

store the current cell address as a variable
replace the cell

channel	1	
topics	0	
last_task		

get_data	{ONERROR _no_vision}{EXEC "VISION C:\QPW\SAMPLES\TRun ObjectVision and load TASKLIST.OVD {INITIATE "VISION","TASKLIST.OVD",channel} {EXECUTE channel,"[@BOTTOM('"'tasks'")]",topics} {REQUEST channel,"Task",last_task} {EXECUTE channel,"[@TOP('"'tasks'")]",topics}	Initiate a conversation with TASKLIST.OVD Go last entry in TASKLIST Store last task to check for end of table Go to first entry in TASKLIST
next_task	{REQUEST channel,"Task",[C(0)R(0)] {REQUEST channel,"Completed",[C(1)R(0)] {IF +[C(0)R(0)=last_task]{BRANCH end_session} {DOWN} {EXECUTE channel,"[@NEXT('"'tasks'")]",topics} {BRANCH next_task}	Store task description in current cell Store task status next to current cell Check if this is the last entry Move cell pointer down a row Make TASKLIST shift to next entry Store next entry in current row
end_session	{REQUEST channel,"Completed",[C(1)R(0)] {EXECUTE channel,"[@APPEXIT]",topics}	Close ObjectVision

help_vision	Cannot run ObjectVision. Please make sure it's in your DOS path, and try again. Press a key to continue	
_no_vision	{MESSAGE help_vision,10,15,0} {QUIT}	

cell_value	Sun	cell_num	5
row_num	10	num_block	A:A1..A:E10

del_negatives	{LET num_block,@PROPERTY("Active_Block.Selection")} {BlockName.Create temp, "+"&[MACROS]num_block} {EditGoto temp} {FOR row_num,0,@ROWS([temp])-1,1,convert_row} {BlockName.Delete temp} {QUIT}	Store coordinates of active block Create a name from the active block Goto the block being checked Check each row of the block Delete the block name Exit the macro
convert_row	{FOR cell_num,0,@COLS([temp])-1,1,check_cell} {RETURN}	Check each cell in the row Try next row
check_cell	{LET cell_value,@CELLINDEX("contents",[temp],cell_num,row_num)} {IF +cell_value<0}{BRANCH erase_cell} {RETURN}	Store cell value Run erase_cell if the cell is negative Return to subroutine convert_row
erase_cell	{BLANK @CELLINDEX("address",[temp],cell_num,row_num)} {RETURN}	Erase the cell, since it is negative Return to the subroutine check_cell

ck
block

jative
ow

ive
:_cell

main_menu	Up	Down	Left	Right	Quit
	Move Up on Enter	Move Down on Enter	Move Left on Enter	Move Right on Enter	Quit this menu
	{LET direction,"{UP}"} {MESSAGE B12..C14,40,12,0}	{LET direction,"{DOWN}"} {MESSAGE B12..C14,40,12,0}	{LET direction,"{LEFT}"} {MESSAGE B12..C14,40,12,0}	{LET direction,"{RIGHT}"} {MESSAGE B12..C14,40,12,0}	{LET direction,"{QUIT}"} {MESSAGE B12..C14,40,12,0}

```
enter_guide {MENUCALL main_menu}
            {MESSAGE B12..C14,40,12,0}
```

```
enter_loop {?}
```

```
~
```

```
direction {RIGHT}
           {BRANCH enter_loop}
```

direction changes based on the user's choice in main_menu. It's not a subroutine.

```
abort_msg  Press Ctrl-Break
           to quit macro
           << Press Any Key To Continue >>
```


1	22	22
---	----	----

33

33

33

33

33

33

cell_type	b	
type_indic	{INDICATE @UPPER(@CELLPOINTER("type"))} {LET cell_type,@cellpointer("type")} {IF cell_type="l"}{INDICATE LABEL} {IF cell_type="b"}{INDICATE BLANK} {IF cell_type="v"}{INDICATE VALUE} {GET next_key} {IF @UPPER(next_key)="{ESC}"}{BRANCH clear_indic}	Set indicator to current cell type Capture the next keystroke Quit if it is Escape
next_key	{Esc} {BRANCH addr_indic}	Execute the keystroke Update indicator, get next key
clear_indic	{INDICATE} {QUIT}	Reset indicator back to its default Exit the macro

```
\m      {LET counter,0}  
again   {; increment counter}  
        {LET counter,counter+1}~  
        {BRANCH again}
```

```
counter      0
```

```
is_done  +COUNTER>=100
```

src_file	c:\temp\commas.txt
curr_line	
file_position	27

make_piped	{GETLABEL "File Name? ",src_file}
	{OPEN +src_file,"M"}{BEEP}{QUIT}
pipe_loop	{GETPOS file_position}
	{READ 1,curr_line}{BRANCH close_file}
	{IF curr_line=","}{replace_comma}
	{BRANCH pipe_loop}
close_file	{CLOSE}
	{QUIT}

replace_comma	{SETPOS file_position}
	{WRITE " "}{}
	{RETURN}

Ask for a file name
Open the file for reading and writing
Keep track of where the macro is in the file
Read a character in the file.
If it's a comma, replace it with a pipe
Read the next character in the file
Move back to the comma that was found
Replace it with a pipe
Return to the main macro

search_string	Fri
\s	{GETLABEL "Search For? ",B1} {Search.Reset} {Search.Block ""} {Search.Find +search_string} {Search.ReplaceBy +search_string}
\n	{Search.Next}

new_name	Nick Piercherd
new_phone	(804) 555-2315

Name: Nick Piercherd Phone: (804) 555-2315

get_record	{WindowShow "MACROS.WB1"} {BLANK E28..E29} {RestrictInput.Enter [MACROS]commeq:D28..F29} {PAUSEMACRO} {IF #NOT#@ISSTRING(E28)}{WindowHide}{RETURN} {IF #NOT#@ISSTRING(E29)}{WindowHide}{RETURN} {LET new_name,E28} {LET new_phone,E29} {WindowHide}	Reveal the notebook so the form can display Clear input form Display Data Restrict Input dialog box Wait for user to finish filling in data If an entry is blank, quit macro If an entry is blank, quit macro Store new name Store new number Hide the notebook again
------------	---	---

width	2current_row	3button_block	A:A1..A:B3
height	3current_cell	2	

make_buttons	{LET button_block,@property("Active_Block.Selection")}	Store acitve block coords
	{BlockName.Create _temporary,+[MACROS]button_block}	Create block name
	{LET width, @COLS([_temporary])}	Store block width
	{LET height, @ROWS([_temporary])}	Store block height
	{FOR current_row,0,height-1,1,button_row}	Process each row
	{BlockName.Delete _temporary}	Delete temp block

button_row	{FOR current_cell,0,width-1,1,create_button}	Process each cell of current row
------------	--	----------------------------------

create_button	{EditGoto _temporary}	Select block
	{EditGoto C(0)R(0)}	Go to upper left cell
	{DOWN current_row}	Move to proper row
	{RIGHT current_cell}	Move to proper cell
	{IF @CELLPOINTER("type")<>"I"}{RETURN}	Check if a label
	{FloatCreate Button,C(0)R(0),0,0,C(1)R(1),0,0,+C(0)R(0)}	Create button

\R {SETMENUBAR}

MENU &Library

&Enter Guide	MACRO [MACROS]enter_guide	Controls which direction select	"No, Yes, No, No
&Delete Negatives	MACRO [MACROS]del_negatives	Removes negative values from	"No, Yes, No, No
D&ates to Labels	MACRO [MACROS]date_lab	Converts dates in active block	"No, Yes, No, No
&Make Pipes	MACRO [MACROS]make_piped	Replaces commas in a file	"No, Yes, No, No
&Quick Search	MACRO [MACROS]S	Perform a quick search	"No, Yes, No, No
&Get Phone Number	MACRO [MACROS]get_record	Gets a name and phone numb	"No, Yes, No, No
Get &Tasks	MACRO [MACROS]get_data	Get tasks from an ObjectVisior	"No, Yes, No, No
Make Macro &Buttons	MACRO [MACROS]make_buttons	Create a button for each label	"No, Yes, No, No
&Cell Typer	MACRO [MACROS]type_indic	Display active cell type on stat	"No, Yes, No, No
&Remove Library	MACRO [MACROS]_remove_macros	Remove this menu	"No, Yes, No, No

add_lib_menu {ADDMENU "/Property",A9..G19}

{SETOBJECTPROPERTY "/Library.Help_Line", "Macros in MACROS.WB1"}

{SETOBJECTPROPERTY "/Library.Depend_On", "No, Yes, No, No, No, No"}

,No,No"

,No,No"

,No,No"

,No,No"

,No,No"

,No,No"

,No,No"

,No,No"

,No,No"

,No,No"