SYMBOX.MBX

Version 1.2

A MapBasic v 500 application by

Jacques Paris

0 General

0 – 1 Purpose

Making visible multiple symbols located at a same point

0 – 2 Installation

Place the SYMBOX.MBX, SYMBOX.INI and SYMBOC_MLC.INI in the same directory.

If SYMBOX.INI is not found by the application, it will generate an new one with some default parameters and no stored schemes in a library.

If SYMBOX_MLC.INI is not found by the application, it will generate a new one in FRENCH only.

0 – 3 Input

A table (base table) containing the reference points and “presence of symbols” data must be open in a mapper.

The “presence of symbols” data must be under the form of “logical” columns, one for each kind of symbol expected in the table.

0 – 4 Output

The symbols existing at a given point are displayed as a horizontal “string” respecting various graphic parameters (see below). They are contained in two new layers: basetable_SYM for the symbols, basetable_BOX for the framing boxes.

Starting with version 1.2, the name of the base table could be truncated in this operation. Mi does not accept names longer than 30 characters; thus, only the first 26 will be kept. This is necessary in order to check for the existence of these associated tables in some operations.

The _BOX is always produced; even if the “box” option is not checked, all the boxes are created invisible to allow for easy editing later on.

Stating with version 1.2, this ..._BOX.tab can also contain the original data corresponding to the “places” for which boxes have been created. (see 1-1)

0 – 5 Processing modes

Given a base table, **the whole table** can be processed, or if a **selection** exists, only the selected points are processed, or the user can identify the points to process **one at the time**. At the end of these operations, the user can choose to **edit** the current results.

An already exiting SYMBOX table (in fact a base table plus the two _SYM and _BOX related layers) can also be reloaded for further processing/editing.

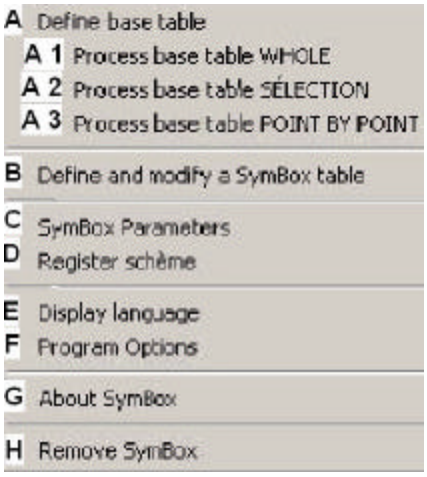
0 – 6 Menu and buttonpads

The application menu is found in the TOOLS menu of the MI mains menu bar for MI=>450; otherwise, it is an menu in the main menu bar.

A toolbar is also created that can be docked or floating (see Program Options).

When a base table as been selected, all the items become accessible as shown in the next two images. Correspondences between menu and toolbar items are shown by letter codes. Not all the items are accessible from the toolbar.

.



During the editing and reprocessing stages, new toolbars appear; they are shown under that topic.

Since version 1.2, the main tool bar contains also an icon (without equivalent in the menu) that cancels a base table definition and the temporary associated tables.

^ the trash can

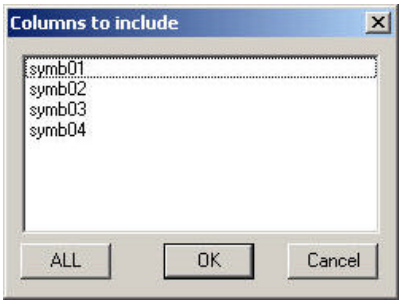
1 Processing a base table

The table must be open in a mapper, alone or with other layers

1 – 1 Define a base table (item and button A)

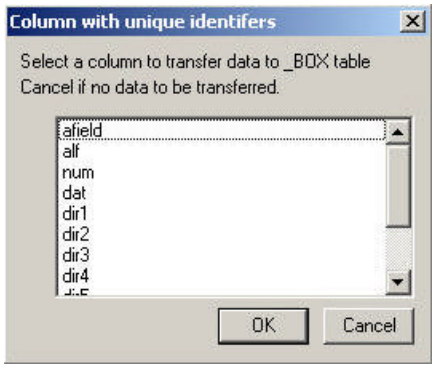
Select the base layer

The program checks if frontwindow is a mapper and if the selected table contains logical columns. If tests are positive



then select the columns “holding” the symbols to be displayed (multiple selection or button ALL); not all the columns must be done at one time, allowing for different presentations from the same base table.

Starting with version 1.2, it is possible to transfer to the result table ..._BOX.tab the original data of the places for which boxes have been created. To specify a transfer, it is sufficient to select in the following dialog a column containing unique identifiers; if no column is selected then “OK”, or if “Cancel”, no transfer will take place.

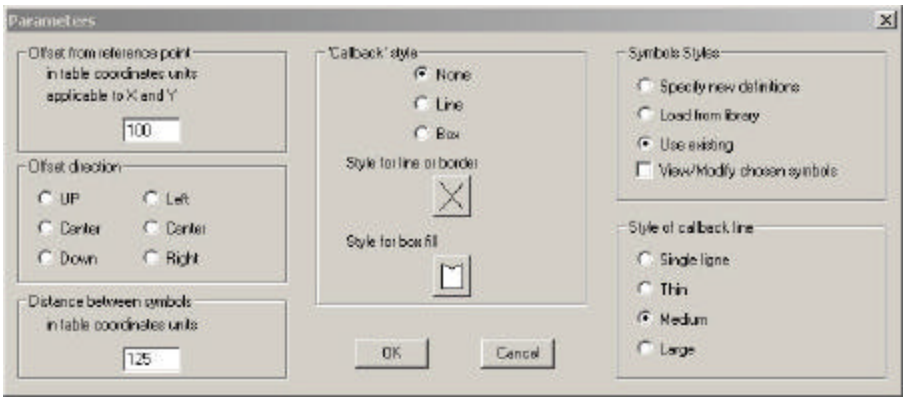


Columns for all types but “logical” are listed. It is preferable not to use decimal/float numeric columns because there might be a loss of decimals in the transfer, and that will make unique identifiers inoperable.

1 – 2 **Symbox parameters** (item and button C)

If you want to choose among the various parameters controlling the final product, you must do it before processing a whole table or a selection.

Access to parameters is only possible when a base table is defined because the symbol definitins are tightly connected with the contents of the table.



Offset from reference point

distance in X and Y between a reference point and the closest corner of the symbol box. Must be expressed in the base table coordinate units.

Offset direction

Combining both controls allows the definition of any of the 9 positions as in the Label requester.

Distance between symbols

As the symbol boxes will be drawn with real world dimensions (the base table coordinate system), spacing between symbols must be specified (in coordinate units) to avoid overlapping; the distance between symbols will vary with point size, shape and map scale.

That measure is also used to build a symbol box: its width is the number of symbols present at that point multiplied by that distance, and its height once that distance. Trials will be required to get the proper adjustment in a specific mapping condition.

“Callback” style

The program provides 3 kinds of “callback boxes” :

None

No visible “box”

Line

A line under- or over- lining the string of symbols with a callback line to the reference point from the closest end of the string or from its middle (horizontally centered).

No lines are drawn for vertically centered symbol strings.
Pen style can be specified with the pen button.

Box

A rectangle frames the symbol string.
Pen style for border and fill style for the box can be specified with the pen and the brush requesters

Style of callback line (bottom right corner of requester)

The symbol box is connected to the reference point by a callback line when the box is not centered horizontally and vertically on the reference point. There are 4 different styles available;

Single line 
Thin 
Medium 
Large 

The “thickness” is always a proportion of the “distance between symbols”’

Symbol Styles

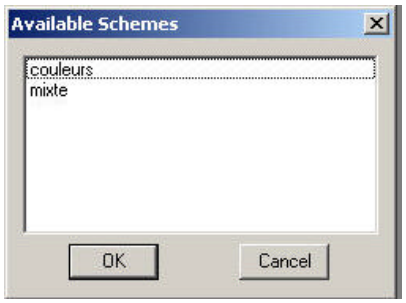
There are 3 ways to specify symbol styles to be used:
NOTE that all the symbols types can be used (MI 3.0, TrueType and Custom-Raster)

“Specify new symbols” will start an iterative process going through each selected column and asking for the style to use

Click on symbol style button, select and accept the new style in the requester, click on OK to move to next column

Or
Click on OK to accept the style and move to next column

"Load from library" uses an existing scheme from a library (see chapter on Scheme Library Management for more details)



Select one of the available schemes then OK

Column names must correspond to the names that are registered in the scheme. If one of the columns in the table does not have a symbol associated to them, the loading will not be complete and processing will continue. However, a scheme can contain more definitions than there are selected columns of a given table.

"Use existing" is used when no redefinition is necessary. It is useful when several maps must be processed with exactly the same scheme.

If used before any definition exists, all symbols will be set to currentsymbol()

In all cases, the "View/Modify Chosen Symbols" check box can be used. If checked, the same procedure as for "Specify new symbols" is started.

1 – 3 Choose a processing mode

Whole table (item and button A1)

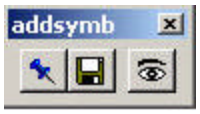
Automatic processing

Selection (item and button A2)

If selection includes points, automatic processing of these points

Point by point (item and button A3)

A small toolbar appears to give the user control over processing



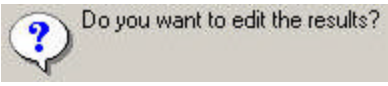
- adding a symbol box:
  activate the symbol button and point to the reference point

- changing parameters:
  use the Eye button. Changing parameters can be done before any addition.

- ending processing:
  use the Diskette button. The same “closing” procedure as described in 1 – 4 will start.

1 – 4 Ending procedure

When processing is completed (automatically or via the Diskette button), the ending procedure begins with



If yes, then a new toolbar appears

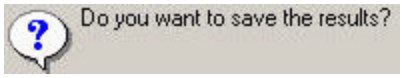
- deleting a symbol box (box even if not visible, callback line if any, symbols)
  click on Scissors button and on the desired “box”

- adding a symbol box
  click on the + button and on the desired reference point

- changing parameters:
  use the Eye button. Changing parameters can be done before any addition.

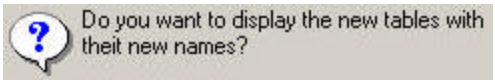
- ending processing:
  use the Diskette button. The same “closing” procedure as described in 1 – 4 will start.

If no



if yes, the results are automatically saved as <basetable name>_SYM..tab and <basetable name>_BOX.tab, placed in the same directory as the base table.

Then



Temporary tables are replaced by the newly created permanent tables in the original mapper. The “future” of these tables is the responsibility of the user.

Closing procedure is completed

If no

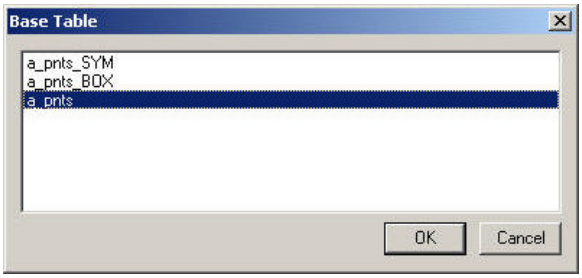
If yes, all temporary tables (and the results) are closed but the base table remains open and "defined" for re-processing.

If no, all temporary tables are closed and the base table is not anymore recognized by this application.

Closing procedure is completed

2 Re-processing a SymBox table (item/button B)

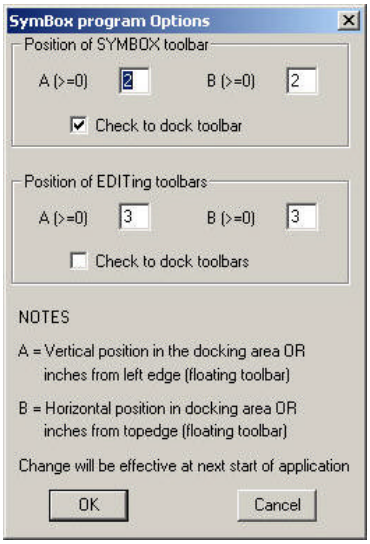
A Symbox table is made of the reference table and the two related layers, ..._SYM and ..._BOX. All these layers must be present in the mapper; _SYM must be the layer on top, _BOX just below it.



Select the base table, not one of the related files. You will have to then select the columns to include as in 1 – 1 A, and use the Edit toolbar as described at the beginning of 1 – 4 Ending procedure.

3 Program Options (item F)

The only option available is the localization of the toolbars. Their “style” (floatting or docked”) and their position can be specified and are recorded in the SYMBOX.INI file for use at the next launching of the application. The main toolbar is set independently from the edit and add toolbars.



4 Managing the Scheme Library

4 – 1 INI file format

A scheme is a set of definitions of column names and symbol styles with the various parameters that define the graphic results. It is stored in the SYMBOX.INI file in a specific format

INI File	Explanations
[General]	Section “general”
…	
schemes="2"	The number of schemes is specified
[Biblio1]	Scheme # 1
…	
[Biblio2]	Scheme # 2
name="mixte"	The name of the scheme
categories="4"	The number of categories in this scheme
offset="200"	Value of the offset (in coordinate units)
dec_hor="3"	Offset direction horizontally (1 left, 2 center, 3 right)
dec_ver="3"	Offset direction vertically (1 top, 2 center, 3 bottom)

intersymb="175"	Distance between symbols (in coordinate units)
style_box="3"	“Box” style (1 none, 2, line, 3 box)
style_line="Pen (1, 2, 0)"	Pen clause for line or border
style_fill="Brush (2, 16777168)"	Brush clause for box fill
style_lang="3"	Callback line style (1 single line, 2 thin, 3 medium, 4 large)
col1="symb01"	Name of column 1
sym1="Symbol (35, 0, 12)"	Symbol clause for column 1
col2="symb02"	… 2
sym2="Symbol (61,16711680,12,"MapInfo Cartographic",1,0) "	… 2
col3="symb03"	… 3
sym3="Symbol (59,255,12,"MapInfo Real Estate",0,0) "	… 3
col4="symb04"	… 4
sym4="Symbol ("CAMP1-32.BMP",0,14,0) "	… 4

The spelling of the headings (what precedes the = sign or the section titles between [ ]) must be respected.

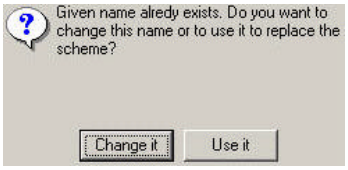
The numbering included in the headings must be continuous starting at 1 and must respect the corresponding definitions (number of schemes, of categories). If the numbering exceeds the set value, the extra lines will be ignored, but if some “numbers” are missing, the program will react badly.

All the “values” are written between “ “.

4 – 2 Assisted management (item D)

Give a new name for a new scheme, or check “replace existing scheme”.

If “newname” already exists, then



“Change it” with return to previous requester, “Use it” will overwrite existing scheme, as with the checked box of the previous requester.

NOTES:

1 - If the number of categories in the newly defined existing scheme is smaller than that in the INI file, the extra “colN” and “symN” lines will not be erased. The user could delete them by hand.

2 – No provision is made for deleting an entire scheme. If the user decides to delete a scheme by hand, he should

- update the number of schemes in the [GENERAL] section
- delete the entire [BIBLIOn] section
- renumber the [BIBLIOx] sections that would be behind the section removed.

5 **Display language** (item E)

SYMBOX is MLC compliant ; on that topic, see “the MLC project” at
www.paris-pc-gis.com/mlc/mcl_main_en.htm

It will accept different languages that are contained in the SYMBOX_MLC.INI file. New languages can be added at will, while respecting the INI file format.

Language change is on the fly. Select another language with the menu option and it becomes the working language for all menus, dialogues and information.

Warning: changing language in the mist of processing may erase some of the choices already made (e.g. the definition of a base table)