MAPBASIC UTILITIES

Michel Larue
SOPAC Secretariat

September 1995 *SOPAC Miscellaneous Report 206*

This project was partly supported with funds provided by the Government of France

TABLE OF CONTENTS

1. INTRODUCTION

1.1 WHAT IS MAPINFO?

MapInfo is the mapping and medium range GIS' software that SOPAC recommends for its member countries. It offers an acceptable compromise between power and user friendliness. As part of a European Union funded project, it has been distributed to ACP member countries. It has been as well adopted by other projects and is becoming a de facto standard in the Pacific region.

One of the reasons it has been adopted is its relative simplicity and the short learning curve associated. The simplicity is at the cost of its power. The successive updates address progressively the limitations, but due to its philosophy, it remains and will remain limited. Some limitations are acceptable, other have to be overcome. Mapbasic offers solutions.

1.2 WHAT IS MAPBASIC?

Mapbasic is a companion product of MapInfo that allows to develop either full stand-alone applications such as EasyMap developed by ERSIS for FFA, or small utilities. It is a development language similar to a Basic language with functionalities to address geographical features.

1.3 ORIGIN OF THE UTILITIES

MapInfo comes off the shelf with some utilities, such as a grid or a scale makers. MapBasic comes also with some more utilities. Along the time, some developers at MapInfo, Kevin Mahonet in particular, wrote utilities to ease their work. Most of those are accessible on INTERNET. Unfortunately, the South Pacific is not yet properly connected to the outside world, as well as interconnected regionally. In addition, they are often designed for a specific use. Some has also been developed at SOPAC. They have been designed primarily by programmers for programmers. Therefore they needed some cleaning before being distributed to end-users.

The present set of utilities has been carefully checked and partially rewritten to be distributed. A user manual is also provided which gives a brief description of what they do, how they work and what they produce. Nevertheless some utilities are sufficiently self explanatory so they do not need to be documented.

The present report makes a summary of what is actually available. It can also be used as a guide of the capabilities of MapInfo/Mapbasic. Therefore, a user in country can seek support to have some extension developed.

As a result it will contribute to a better use of the tool installed.

These utilities are full standalone programmes but each of them performs one single function. They also give indications of ways and mean of programming with MapBasic.

Geographical Information system

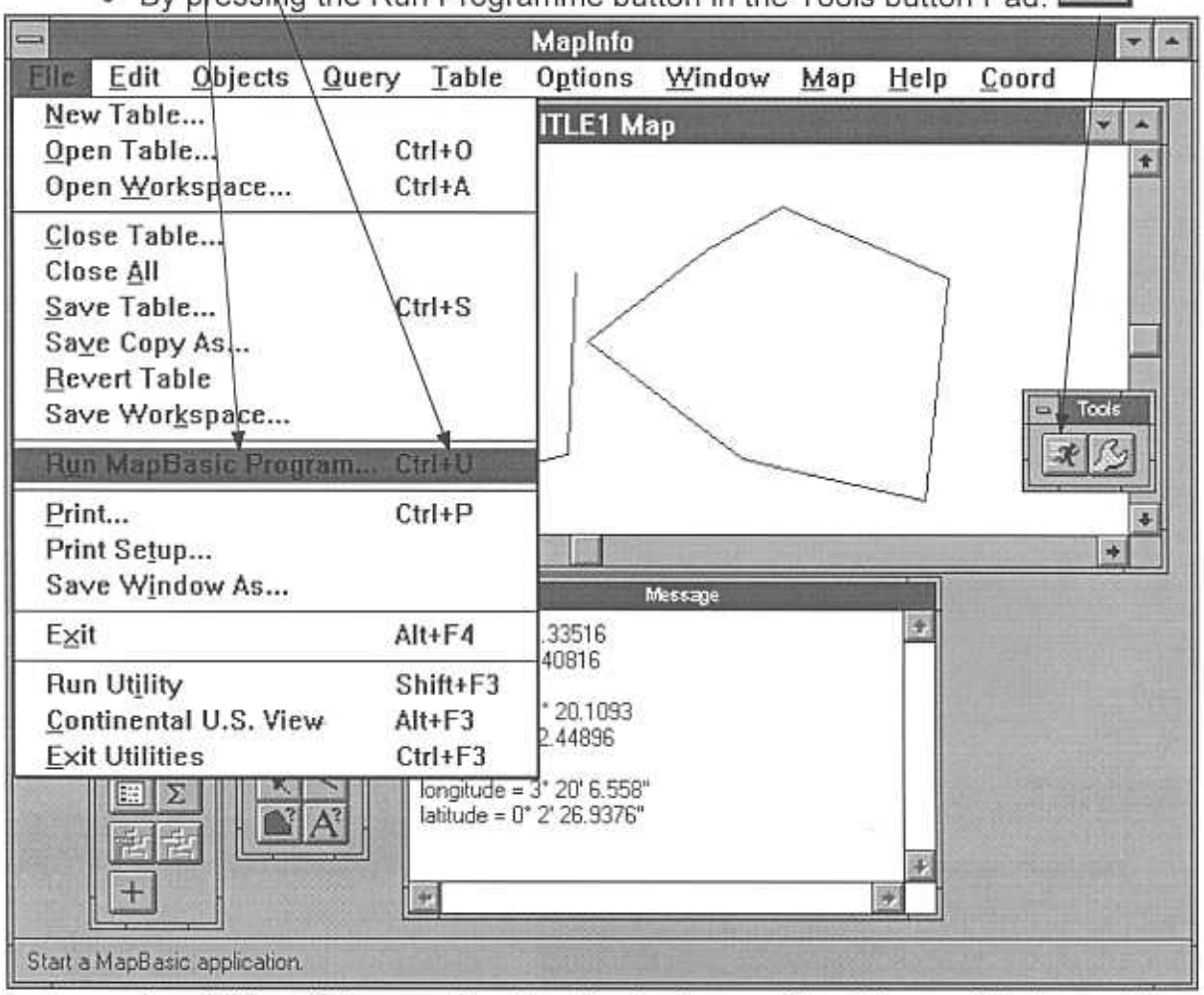
2 ACP, Africa, Carribean, and Pacific, countries which signed the Lome covention and can receive funds from European Union

1.4 How to launch a MapBasic programme?

Mapbasic applications should not be confused with the MapBasic Window in MapInfo. The Mapbasic Window is a Window inside MapInfo that records most of the commands that the user performs from the menu commands as macro that can be stored and invoked to perform similar action. The MapBasic window is opened either by cliquing on the button or by `Menu > Options > Show MapBasic window`. For more information on MapBasic windows, see MapInfo reference manual p. 294.
From now on when we will refer to Mapbasic, it will mean a Mapbasic application. Mapbasic executable applications are files with a .MBX extension. Source files are stored with a .MB extension. Executable MBX files need MapInfo to be executed, they are not stand-alone applications.

There are several ways to launch a MapBasic programme.

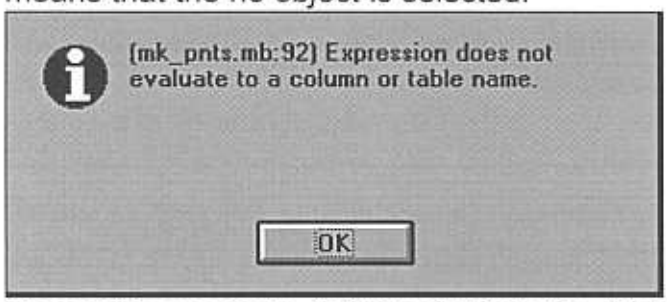
- Ctrl+U
- Menu File > RUN MAPBASIC PROGRAMME
- By pressing the Run Programme button in the Tools button Pad: 



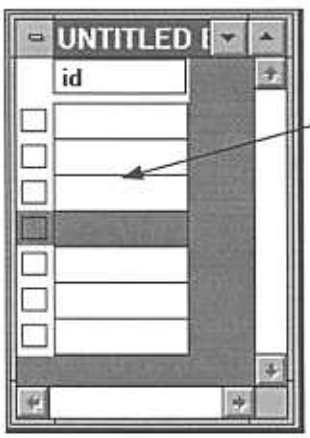
- In addition, if the user has the Mapbasic compiler, it is possible to run the application from MapBasic after having compiled the project.

1.5 GENERAL USE OF THE APPLICATIONS

This applies at least for the utilities developed at SOPAC. It adopts an ***Object-Verb*** syntax. The same way as in a word processor the user selects a text (Object) and then apply an action such as Bold (Verb). Many utilities are applied on the selection that has to be made first. Any way to make the selection is acceptable: use of the Select Button  , the Marquee Select Button  , of a SQL (`Menu>Query>SQL Select`). If the selection is made from the Browser, and the utility a mapping one, a Mapping window has to be made active before launching the utility. If this is not the case, a message such as the following is issued. It is not quite explicit. In fact it means that the no *object* is selected.

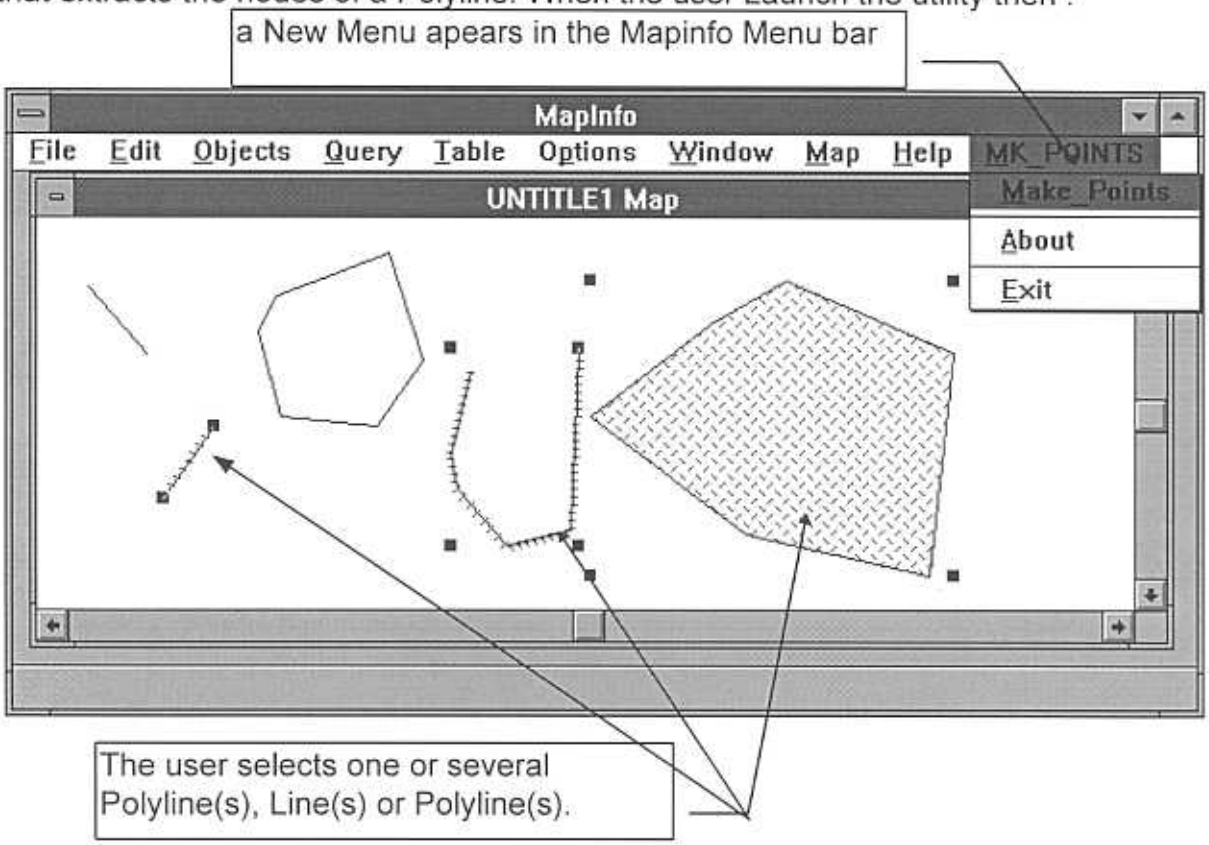


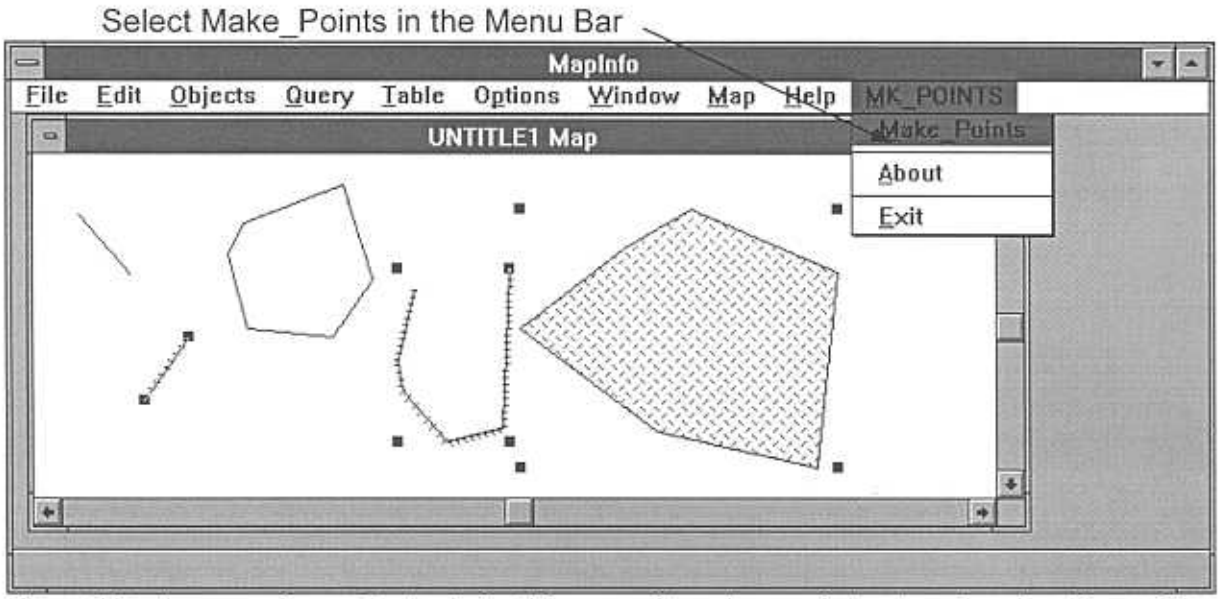
The same message is issued when the table is not Packed, lines in the table are empty. This happens when objects have been deleted from a map or lines deleted from the browser. When the browser is opened, some lines, corresponding to the deleted object appear greyed. To remove these lines, the user has to:



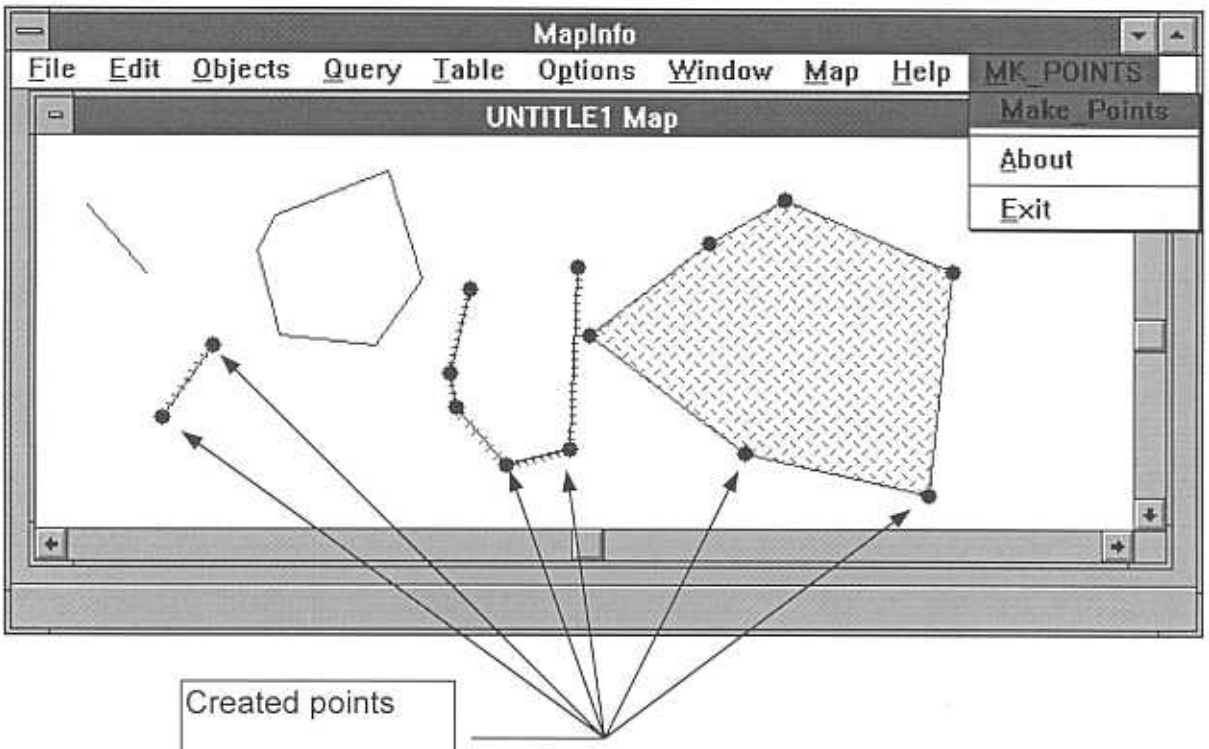
`Menu> Table > Maintenance > Pack Table...`

To explain to the user how to use them, we will take one example: `MAKE POINTS` that extracts the nodes of a Polyline. When the user Launch the utility then :





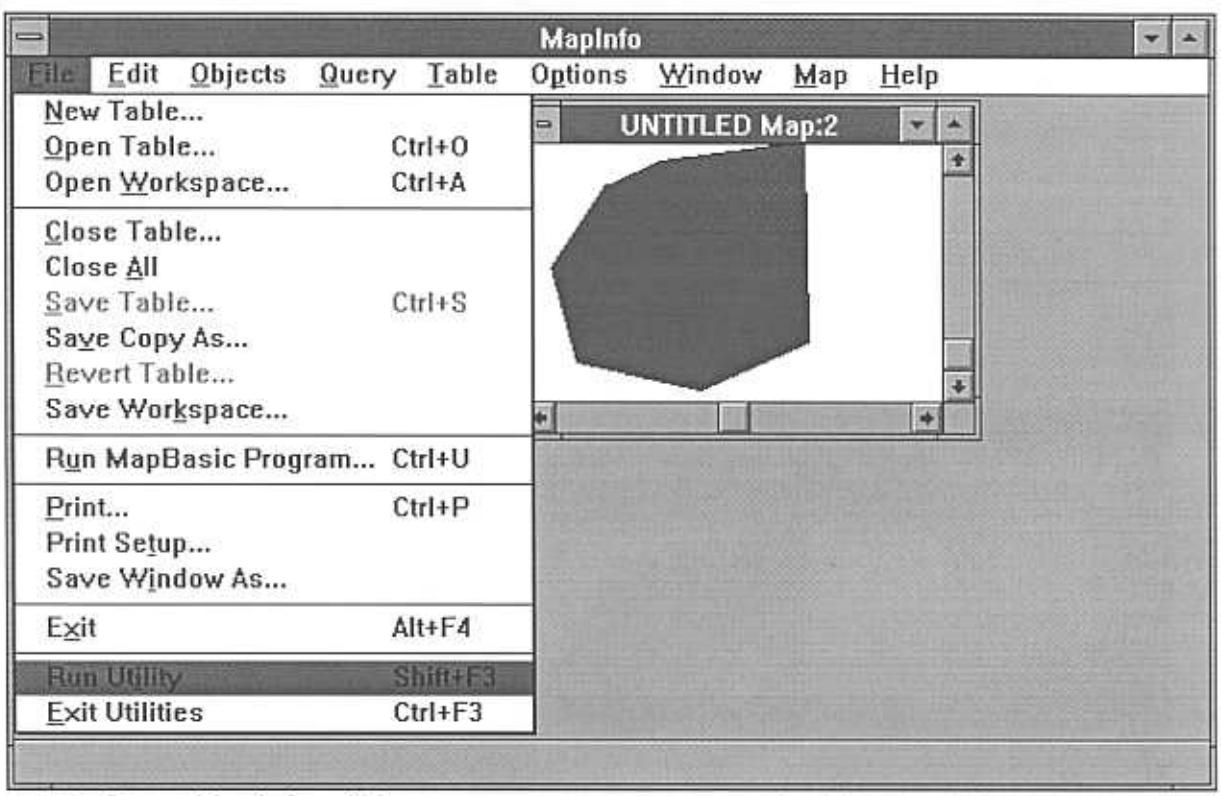
The utility then performs its task; in this case it makes points at nodes location of the selected objects.



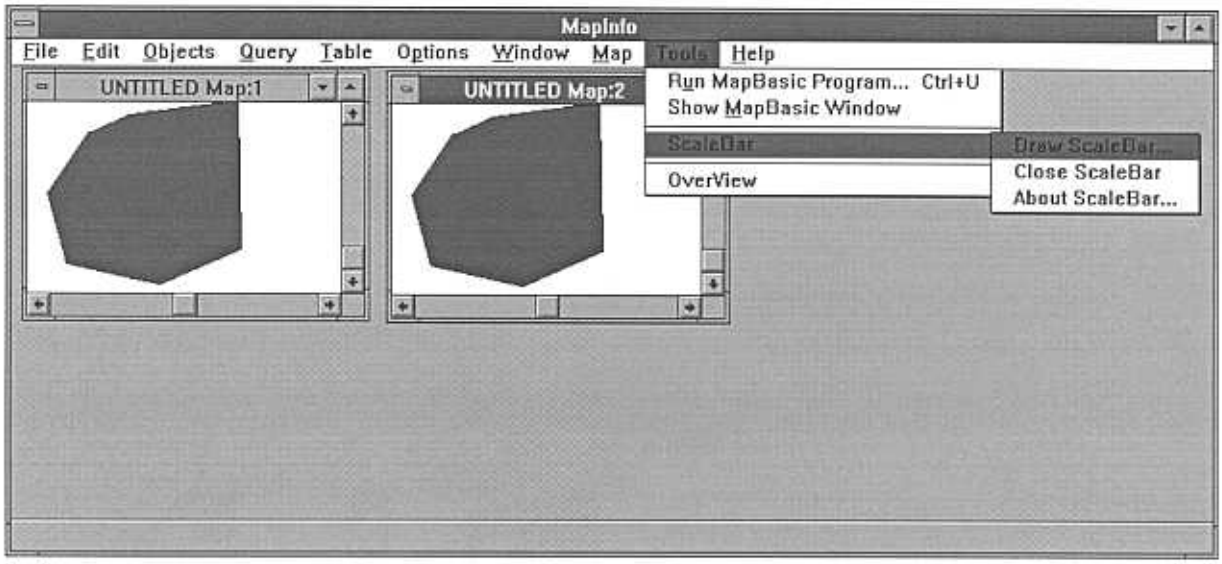
1.6 OTHER WAYS

There are four ways a utility can behave.

1. It can create a new menu in MapInfo menu bar, like in the example above.
2. It can act directly without any warning nor intermediate step.
3. It can add a menu somewhere else in MapInfo menu system.



4. Some MapInfo utilities create a `Tool Menu` and add new utilities developed with the same options in this submenu



1.7 STOP A MAPBASIC PROGRAM

Utilities either leave MapInfo when they have performed their task or remain *asleep*. Most of them have an `exit` option, as well as an `about`, which gives a brief description of the task performed.

If an error occurs, the utility usually leaves MapInfo. It has to be launched again.

1.8 COSMETIC LAYER

Another general feature is that unless other mention, the result of the operation is put into the Cosmetic layer. The utilities work the same way as the `Draw Autolabel` feature in the standard MapInfo (See MapInfo Manual p. 97). The user must be **very careful** that **MapInfo does not prompt before erasing the cosmetic layer**. If the user closes the window, then the content of the cosmetic layer will be lost. See MapInfo manual p. 30 for Clear Cosmetic objects Command and p. 273 for Save Cosmetic Objects Command). When objects are transferred in the cosmetic layers, the table structure or its content is not transferred as well.

1.9 ADDITIONAL FILES REQUIRED

Some utilities require to have additional files such as a special symbol set. Whenever necessary, the standard MapInfo set is saved under the name `BACKUP.FNT` and restored when exiting the utility. The copying and saving of the FNT files can be done as well outside the utility. Copying the file can be done in the file manager of windows. It is advised to keep a copy of the standard `MAPINFO.FNT` file, either the original one or the one that the user chose to work with. Then copying (alternately renaming) the customised symbols file to `MAPINFO.FNT`. Then, in the MapBasic window of MapInfo type

```
Reload symbols.
```

This makes the new symbol set active.

1.10 DISTRIBUTION FILES

This report comes with a floppy disk. It includes the utilities, sources and executable files, customised Symbols, and some tables set for a standard installation.

2. LIST OF THE UTILITIES

As already pointed out, the utilities are coming from three main sources: SOPAC, MapInfo as standard distribution and Kevin Mahoney (referenced as KGM) utilities.

2.1 SOPAC UTILITIES

- COORDINATES
- GRID5
- JOINT POINTS
- MAKEPOINTS
- READBITMAP
- RESIZE WINDOW
- PRINT ACTIVE SYMBOLS
- VECTOR COMPONENTS
- ADD ARROWS TO VECTOR
- EXPORT NODES
- BREAK REGIONS
- IMPORT MULTIPLE MIF
- IDENTICAL WINDOWS
- OPEN TABLE (FULL PATH)

2.2 KGM UTILITIES

- AGGREGATE LINES
- AGGREGATE SYMBOLS
- ASCII EXPORT
- BUFFER SELECTION
- CLONE WINDOW
- CLOSE TABLE(S)
- COMBINE LAYERS
- CONNECT POINTS
- CUT STREETINFO FILES
- CUT STREETS IN REGION
- DDMMSS TO DECIMAL DEGREES
- DISTANCE TO OBJECTS
- DRAW LINES TO POINT
- FIND STREET INTERSECTION
- FIXED LENGTH ASCII IMPORT
- INTERSECTION OF 2 REGIONS
- LABEL STREETS AUTOMATICALLY
- LINEWITHIN
- MAKE SCREEN GRID
- OPEN MULTIPLE FILES
- OPTIMISE LOCATION
- OPTIMISE TABLE STRUCTURE
- POINT DISPERSION
- PROJECTIONS, ETC.
- RANDOM NUMBER GENERATOR
- REGION WITHIN
- RESIZE MAPPER FOR LAYOUT

- SELECT ALL IN SCREEN
- SHADELAYER
- SORTTABLE
- STREET ALIASES
- STREET NAMES WITHIN
- UPDATE COLUMN VALUES
- UTILITY (Launch a utility with Full path)

2.3 MAPINFO

- AUTOCAD LINK
- ADD NODES
- GEOCODE
- GEOSCAN
- MAKESHIELD
- SYMBOL EDITOR
- INFOTABLE
- DDEEXAMPLE
- SCALEBAR
- OVERVIEW
- TEXTBOX

3. USER MANUAL

3.1 UTILITY

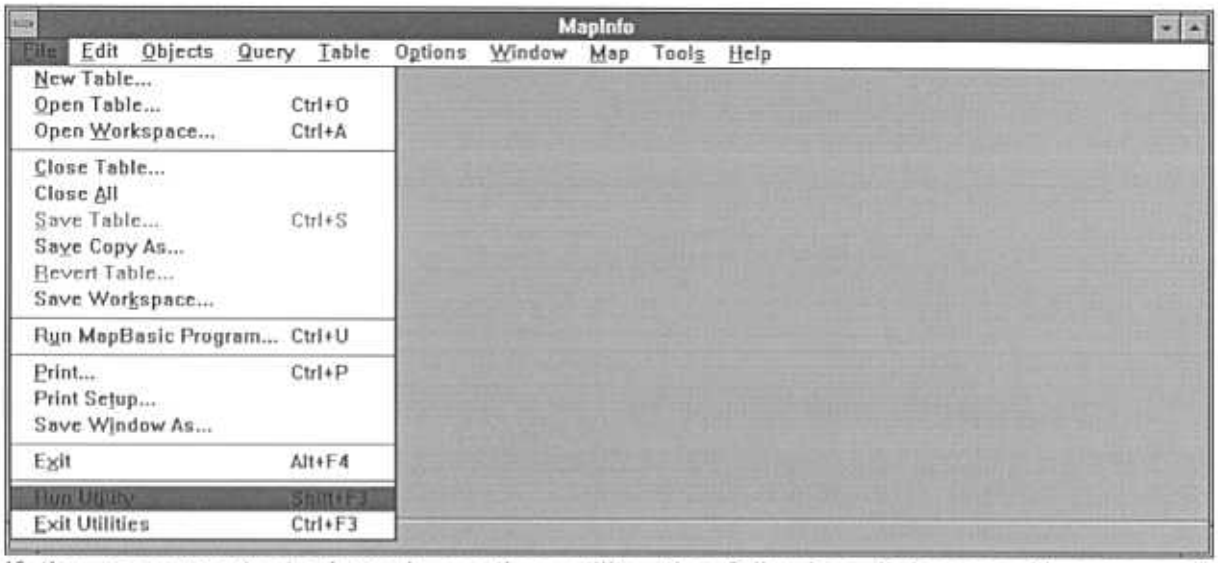
3.1.1 File Name
UTILITY.MBX

3.1.2 Use
Before we start, there is a useful utility called "UTILITY" that can be used to help handling of the utilities. One of the difficulties in Windows is to find where a particular file is in the sub-directory tree. Windows does remember one particular location and whenever an open file is performed, the dialogue will be issued from this particular location. It would be very handy to be able to store the full path to a particular file, in this case a utility. This program uses one capability of MapInfo that is: it is a data base. Usually it is used for its geographical data, but it can handle tabular data as well.

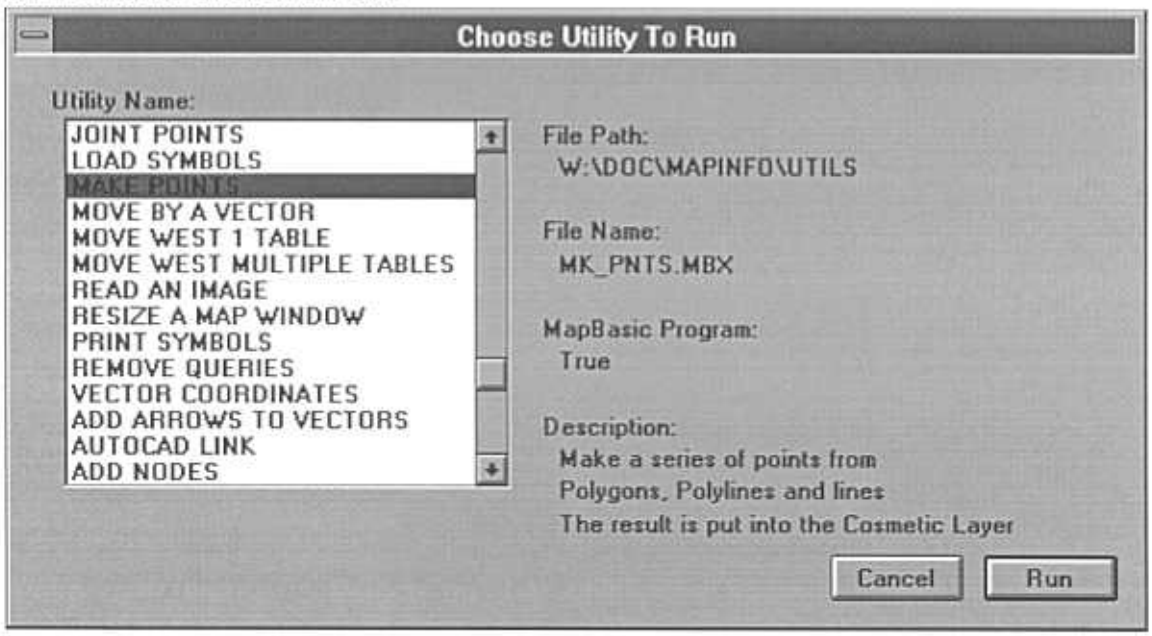
This utility makes use of a table with the following structure:

Field	Field Name	Type	Width
1	UTIL_NAME	Character	25
2	FILE NAME	Character	12
3	FILE-PATH	Character	60
4	MAPBASIC	Logical	1
5	DESC1	Character	60
6	DESC2	Character	60
7	DESC3	Character	60
8	FILES_NEED	Character	60
9	ORIGIN	Character	10
10	USE	Memo	10
11	BUGS_LIMIT	Memo	10
12	NEXT	Memo	10

When launched, the utility adds a menu in the file system and opens this database in a hidden state. Therefore the user cannot browse or edit it. To do so it is necessary to open the table separately.



If the user wants to launch another utility, the following dialogue will appear. It presents the contain of the data base, a full name of the application, its full path and a brief description. To actually launch the application, the user just either double click on the name of click on OK.



Although a version of UTILITY database is distributed, it will have to be edited by the user to fit the configuration on the computer. The path will have to be modified to be most probably C:\MAPINFO\UTIL\.
The logical field MapBasic Program allows to launch as well true DOS/Windows executable from the same dialogue box.
If the database is a .DBF file then it can be read from ACCESS.

3.1.3 Possible developments

1. An automatic update of the database
2. To pass parameters to applications

3.2 OPEN TABLE

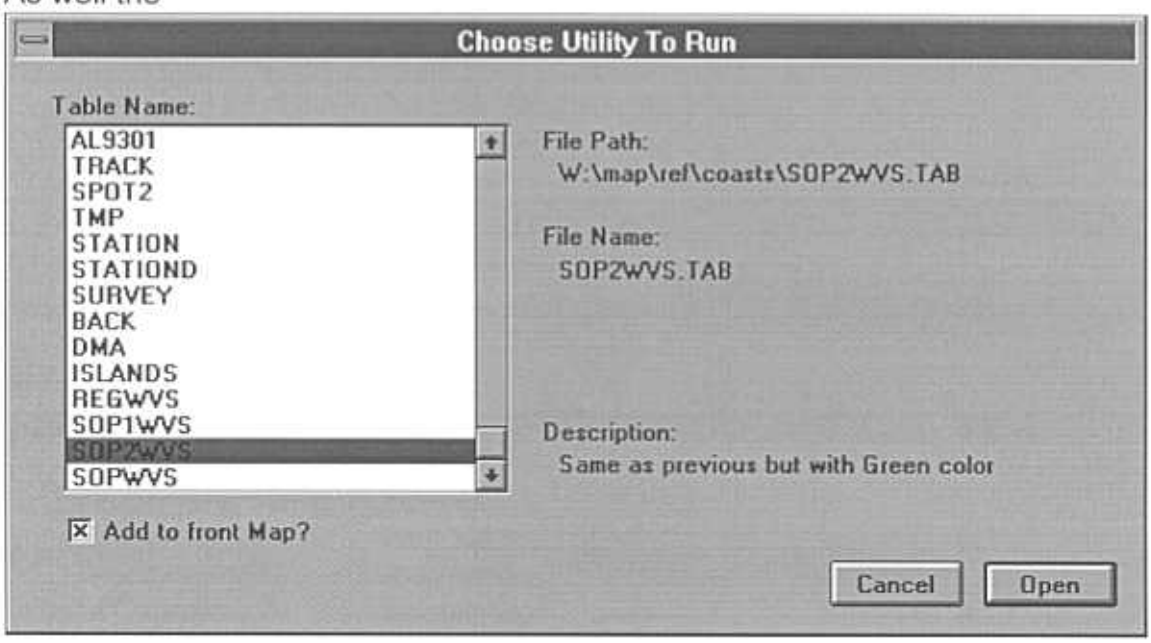
3.2.1 File Name

OPEN_TAB

A similar utility, OPEN_TAB makes use of the same capability of MapInfo to handle full path. It makes use of the table DATA_ALL.
The structure is :

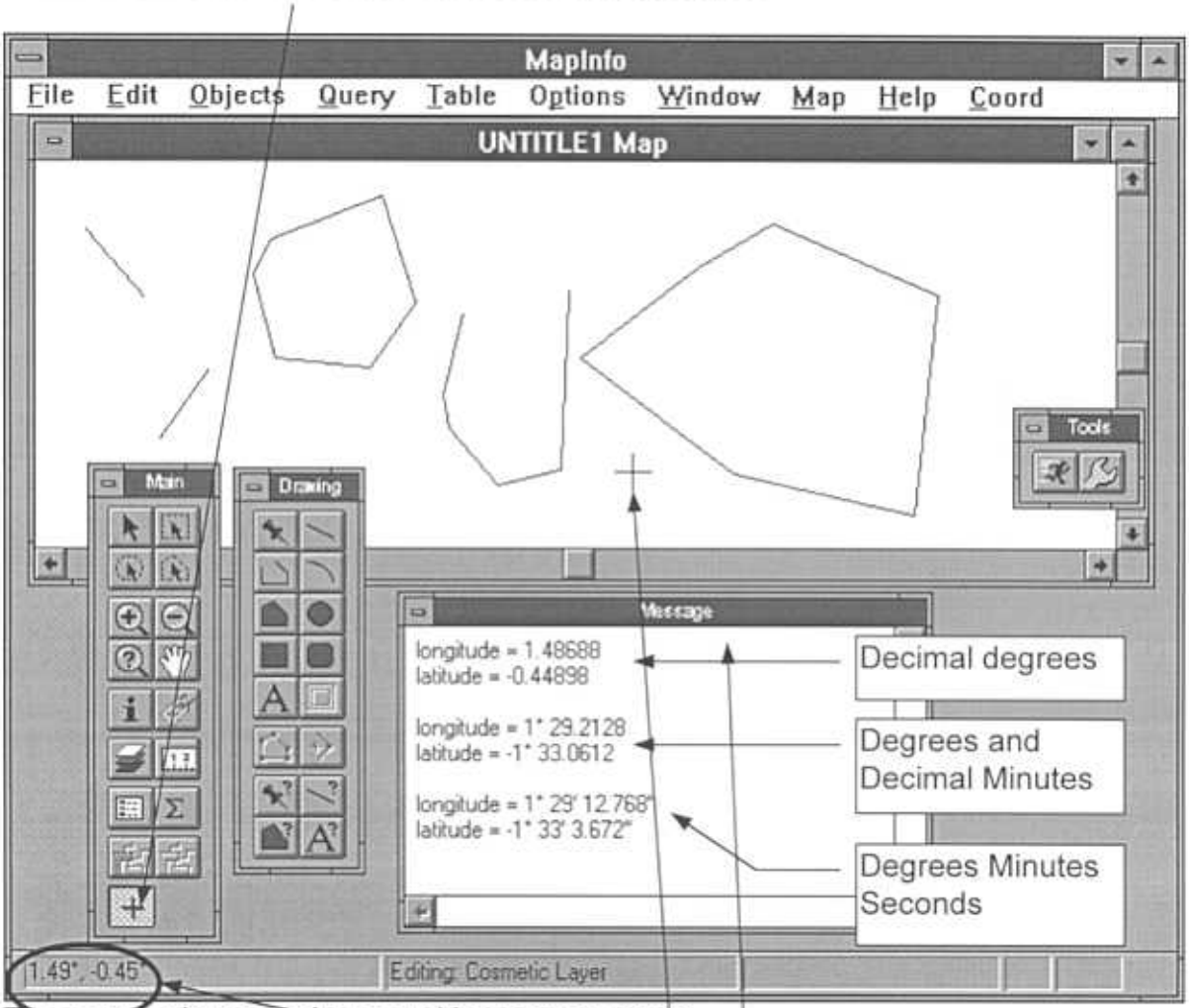
```
Field   Field Name   Type        Width
    1   TABLE_NAME   Character      25
    2   FILE_NAME    Character      12
    3   FILE_PATH    Character      60
    4   DESC1        Character      60
    5   DESC2        Character      60
    6   DESC3        Character      60
```

As well the



COORD

It prints various forms of the coordinates of a point. When launched, the utility makes a new Tool Picker button active in the main button pad.



The cursor changes shape and becomes a cross

When clicked, a message windows made active by the Print command opens and the position of the point is printed. It enriches the standard MapInfo display.

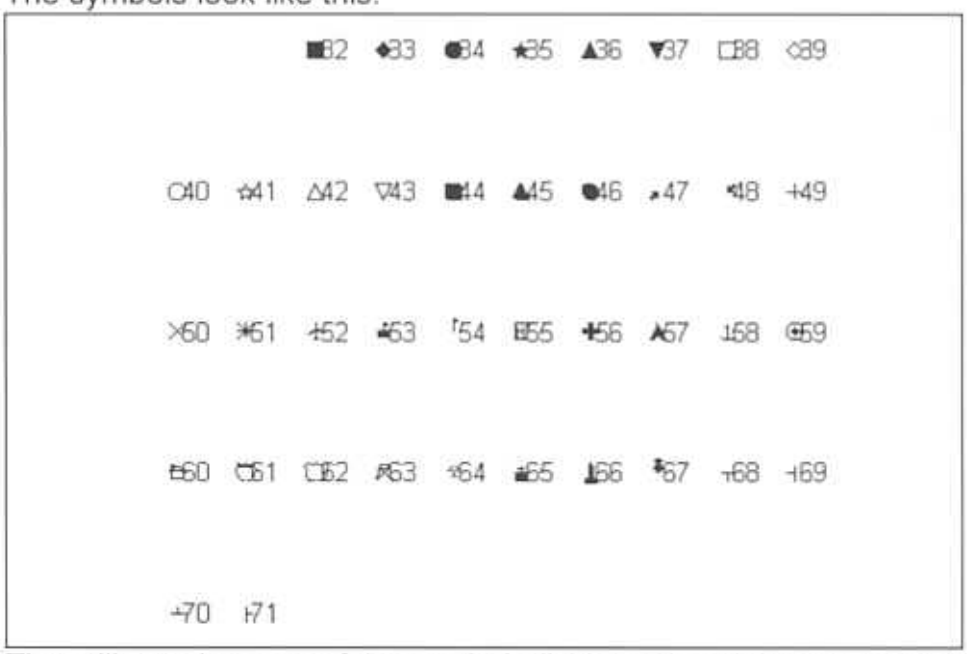
3.3 GRID5

3.3.1 Filename
GRID5.MBX

3.3.2 Purpose:
Create a customised Grid:

3.3.3 Origin:
It is a SOPAC modified version of the GRID received with MAPINFO.

3.3.4 Additional files required
It makes use of a customised set of symbols, called `GRID.FNT`. Before launching the utility, make sure that the file is in your MapInfo directory (most likely C:\MAPINFO). The symbols look like this:

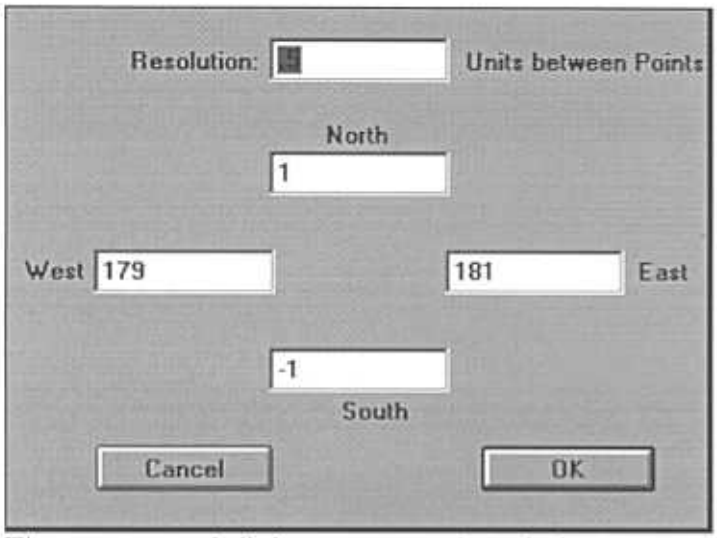


The utility makes use of the symbols # 68 to 71 and the symbol # 49. This drawing of the symbol set has been produced by a utility called TST_SYMB, see under this name.

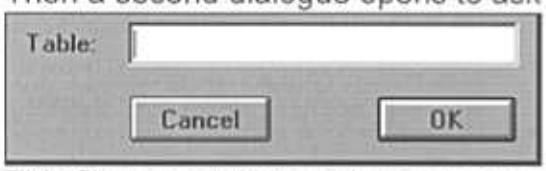
The main differences between this utility and the standard one distributed with MapInfo are:

- It places crosses instead of continuous lines
- It automatically draw text around the map
- It allows any projection system
- It allows to use any unit available under MapInfo

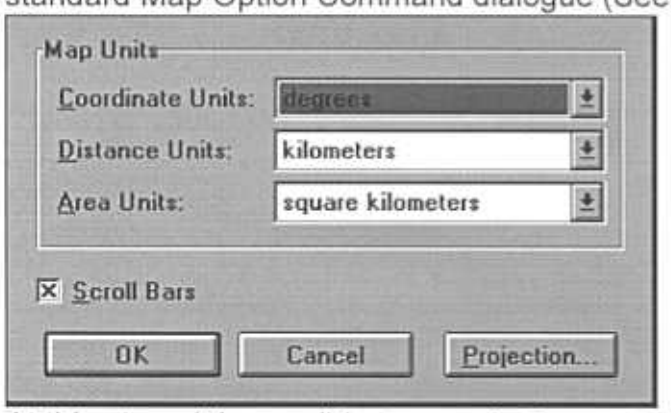
When launched, A Dialogue appears. At this time the user has to know which unit will be used.



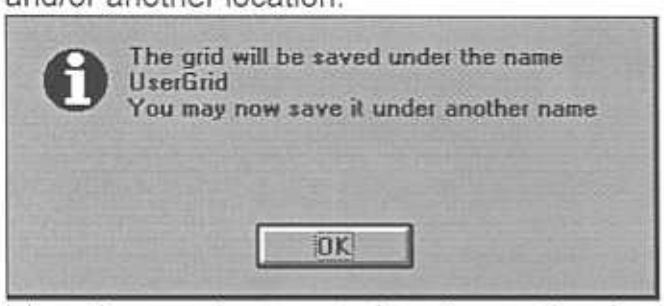
Then a second dialogue opens to ask the name of the file to create.



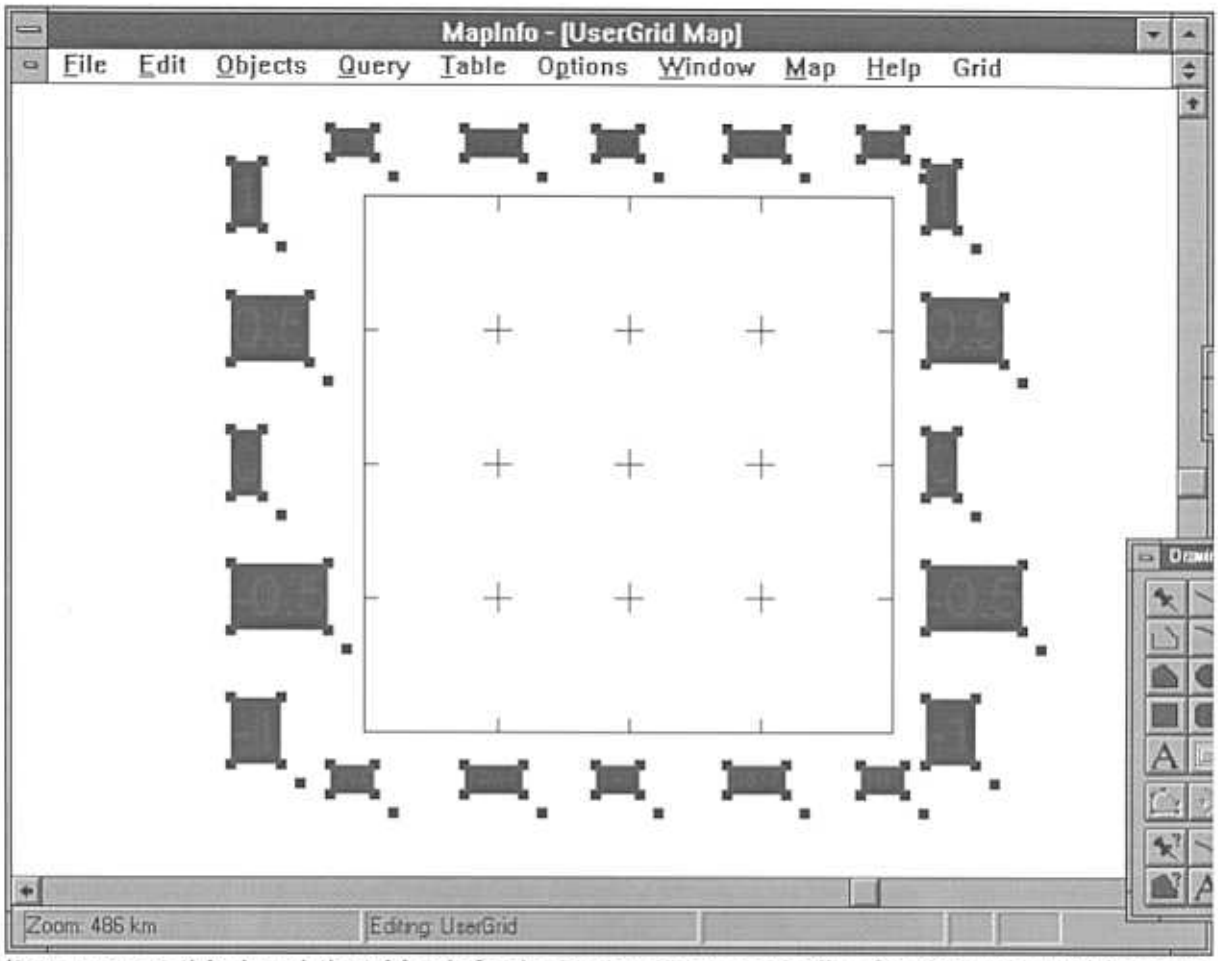
This file name will be only temporary, as it is automatically placed in the sub-directory where the utility has been launched from.
A third dialogue appears to set the projection system of the grid to create. It is the standard Map Option Command dialogue (See MapInfo manual p. 215).



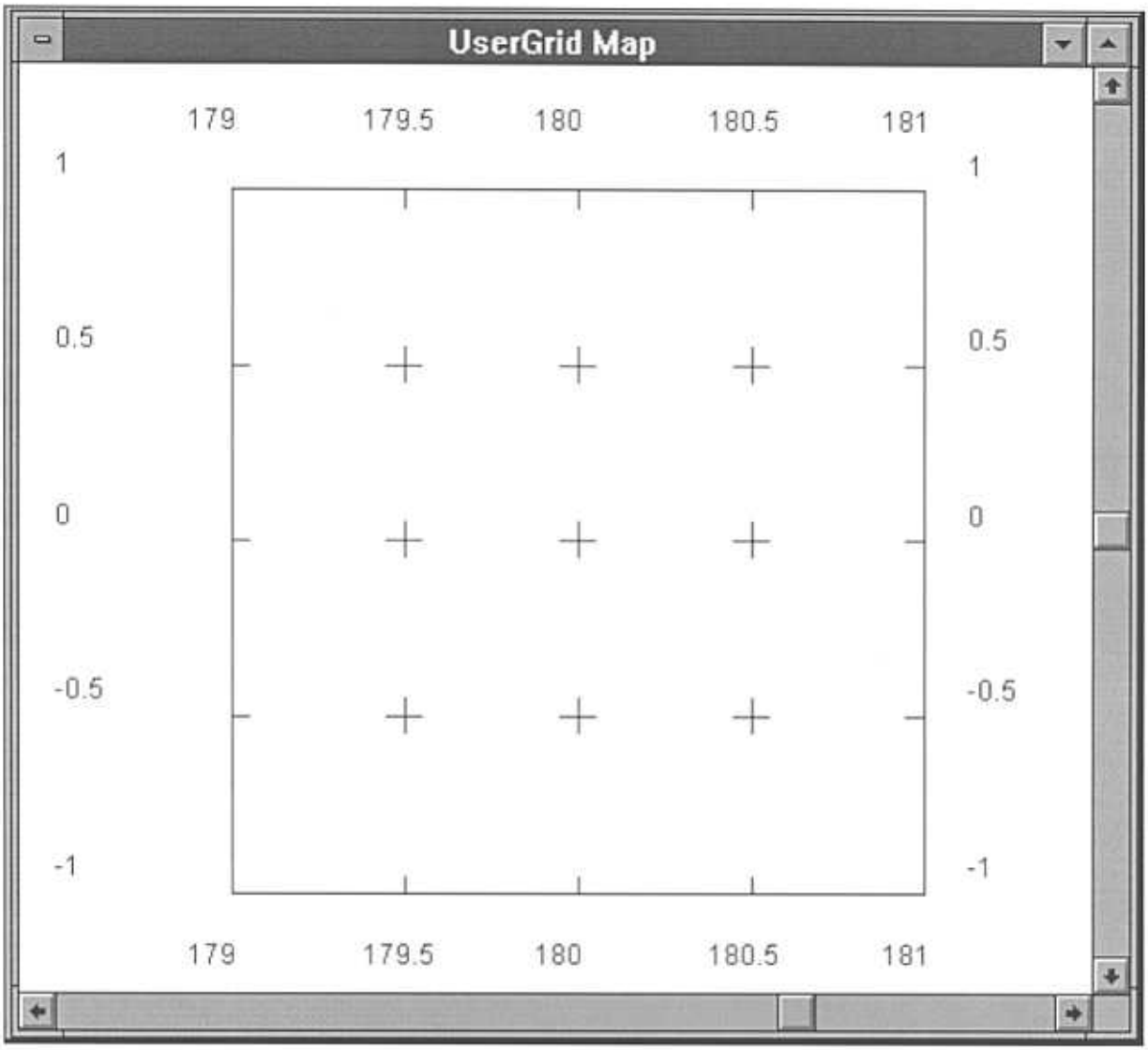
At this stage it is possible to completely customise the grid to be created.
Then when the grid is created, the user is prompted to save it under another name, and/or another location.



Then the user is prompted to change the size and font of the text.



It seems at this level that MapInfo does not react normally. So the user will have to click to the font change button  again.
After re-sizing the window, the result looks like this:


UserGrid Map
179
179.5
180
180.5
181
1
0.5
0
-0.5
-1

3.4 JOINT POINTS

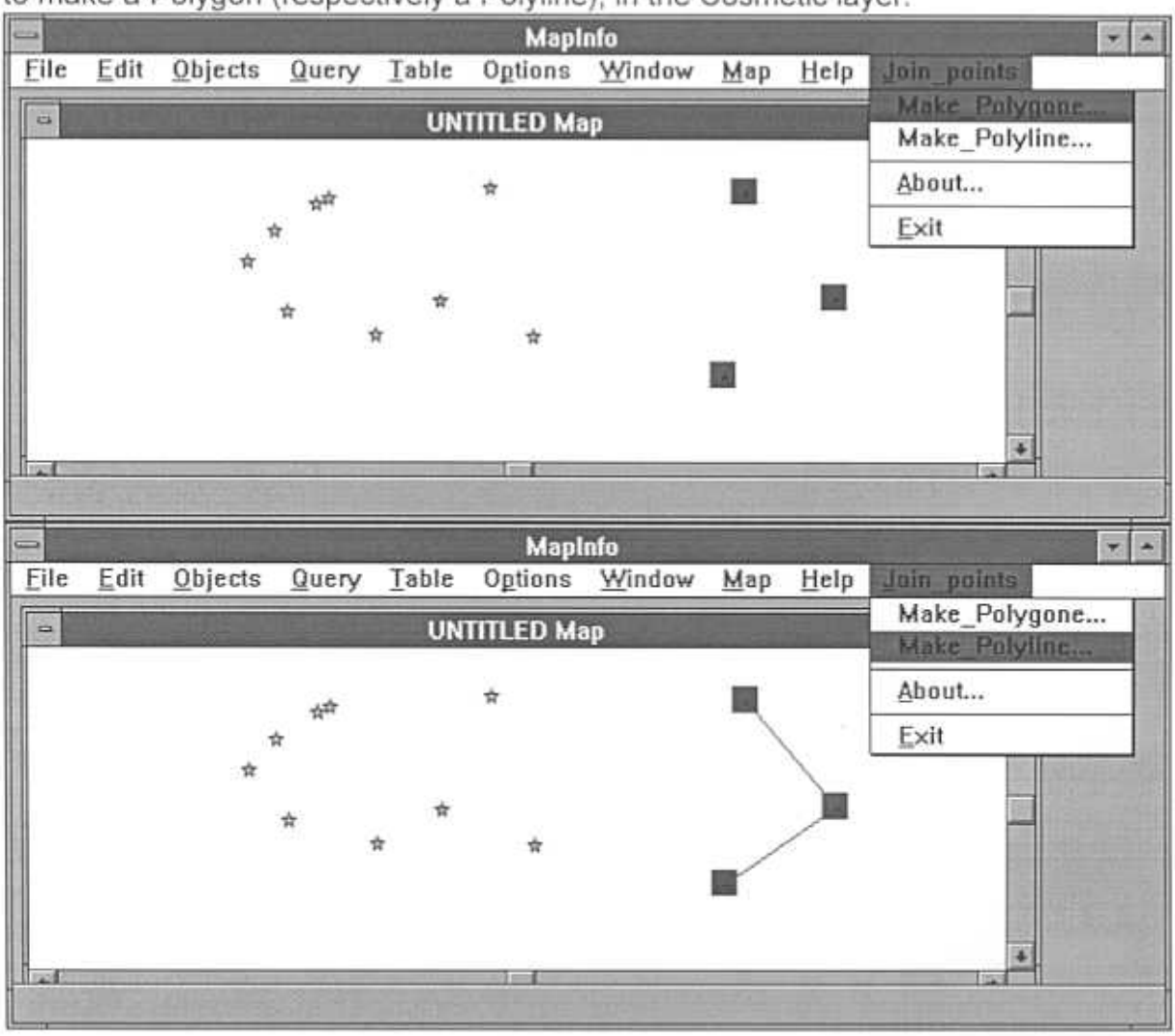
3.4.1 FILENAME
JN_PNTS

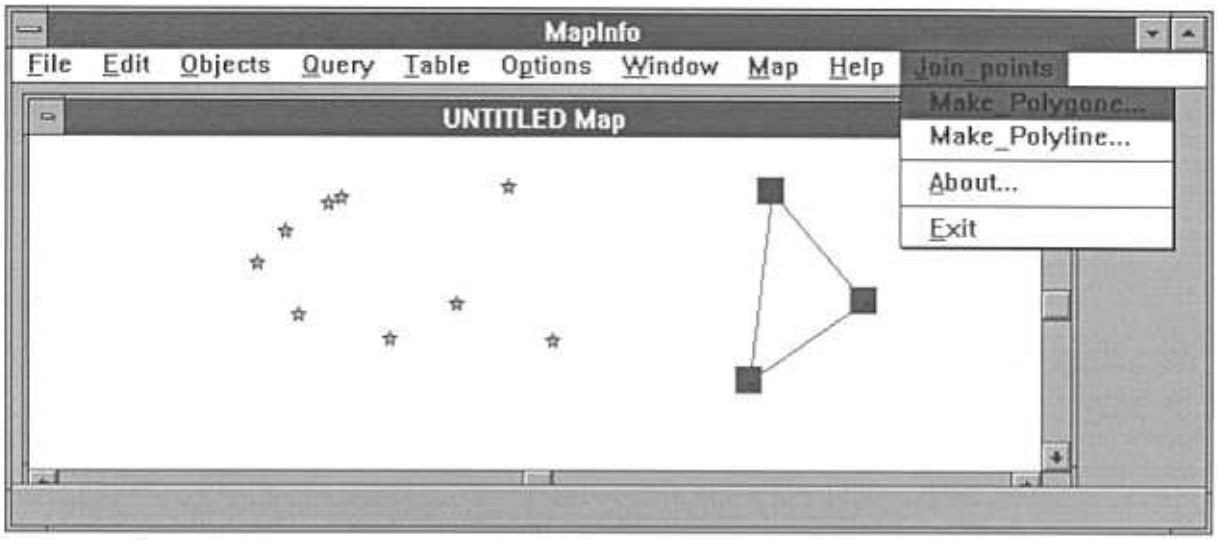
3.4.2 Purpose:
Join selected points to form either a Polyline or a Polygon. The result is put in the Cosmetic layer that becomes editable during the process. It is the reverse function of MAKE POINTS.

3.4.3 Origin:
Developped at SOPAC

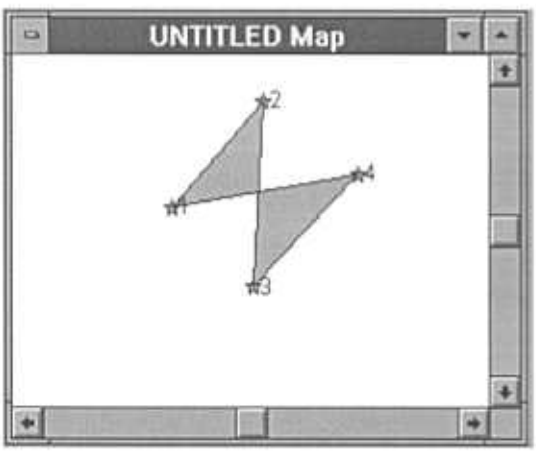
3.4.4 Additional files required
None

3.4.5 Example:
Starting situation: A series of points in a table, here untitled. All or some of them are selected. Then by choosing either `Make_Polygone` or `Make_Polyline`, it possible to make a Polygon (respectively a Polyline), in the Cosmetic layer.



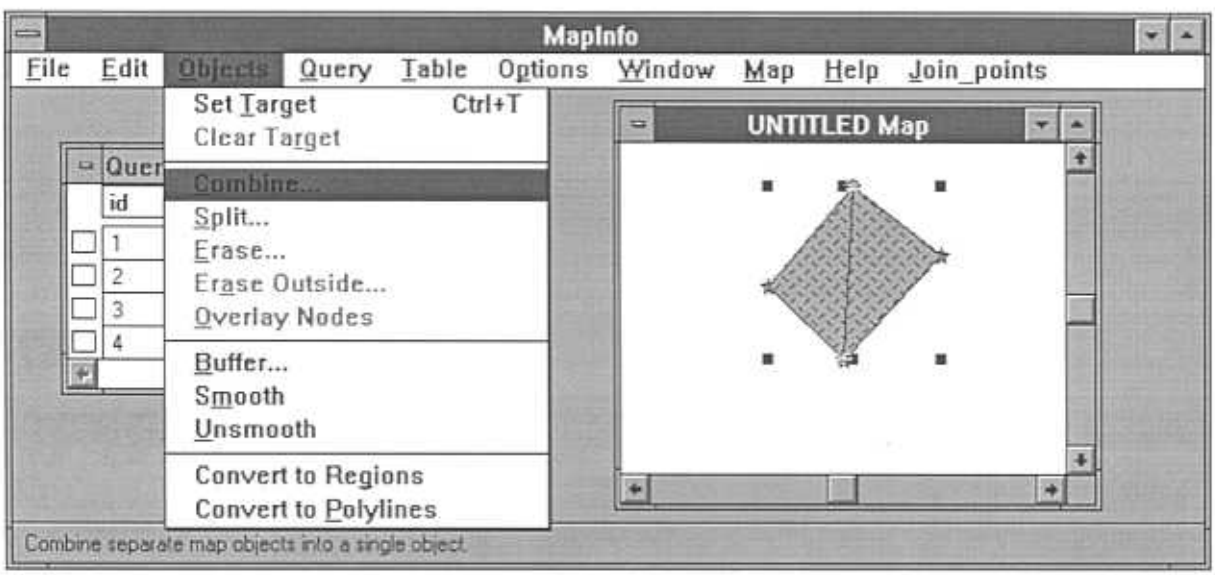


3.4.6 Known bugs, limitations



The utility takes the points in the order they are in the table, not in the selection. Therefore one can end with a situation that looks like the following.

The way to overcome this is to create two adjacent triangles and then combine them.



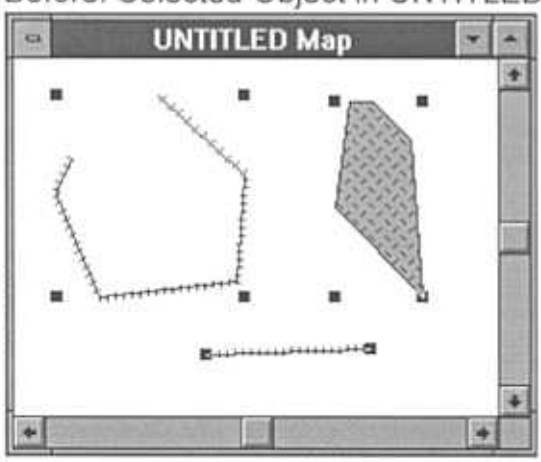
3.5 MAKE POINTS

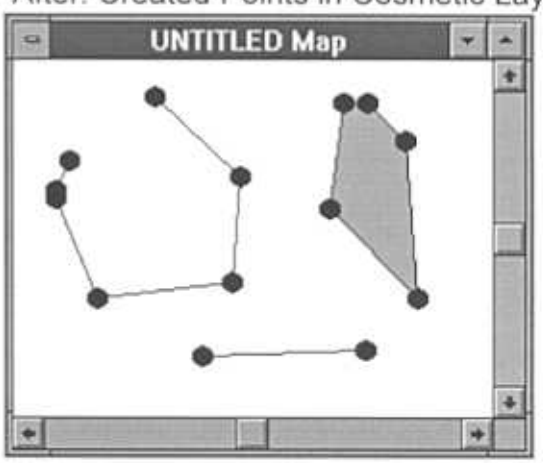
3.5.1 FILENAME
MK_PNTS

3.5.2 Purpose:
It extracts Points from the nodes of Polygons, Polylines and Lines. It is the reverse of JN_PNTS

3.5.3 Origin:
Developped at SOPAC

3.5.4 Additional required files:
No Symbols necessary
Example
Before: Selected Object in UNTITLED. After: Created Points in Cosmetic Layer





3.5.5 Known Bugs, Limitations:
It works only with Polyline, Polygon and Line, not rectangles.

3.6 READ A BITMAP

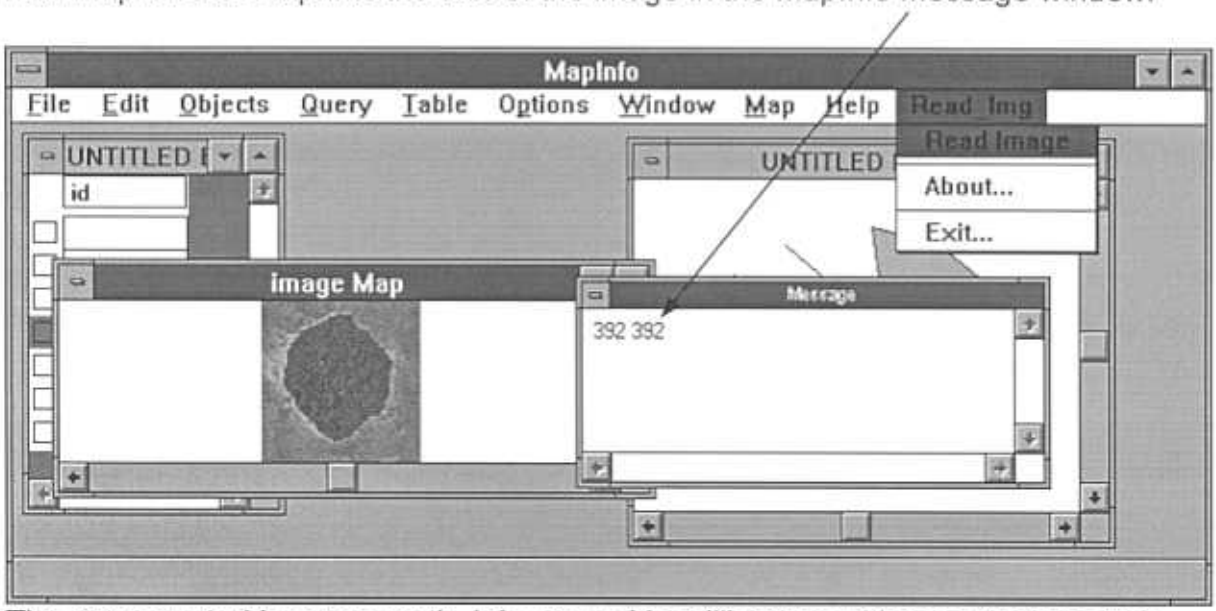
3.6.1 Filename
RD_BMP

3.6.2 Purpose:
The utility reads an image saved as a Windows Bit Map file with a BMP extension and display it in a separate window.

3.6.3 Origin:
Developped at SOPAC

3.6.4 Additional files required
No Symbols necessary

3.6.5 Usage
When Launched, the utility prompts the user to open a BMP file, then it displays it. To do that, it reads the size of the file, then creates a `.TAB` file. It then opens the new map created. It prints the size of the image in the MapInfo Message window.



The map created is a non earth. It is created in millimetres with one Pixel per mm.
Example of a .TAB file.

```
!TABLE
!VERSION 300
!CHARSET WindowsLatin1
Definition Table
File "D:\DATA\SAVO\IMAGE\SAVO3.BMP"
Type "RASTER"
   (0,0)           (0,0)       Label "Pt 1",
   (392,0)         (392,0)     Label "Pt 2",
   (0,392)         (0,392)     Label "Pt 3",
   (392,392)       (392,392)   Label "Pt 4"
CoordSys NonEarth Units "mm"
Units "mm"
```

3.6.6 Known bugs, limitations

It replaces an existing file with the same name without prompting. It opens the image as IMAGE Table, if an image of the same name exists, there could be a conflict.

3.7 RESIZE

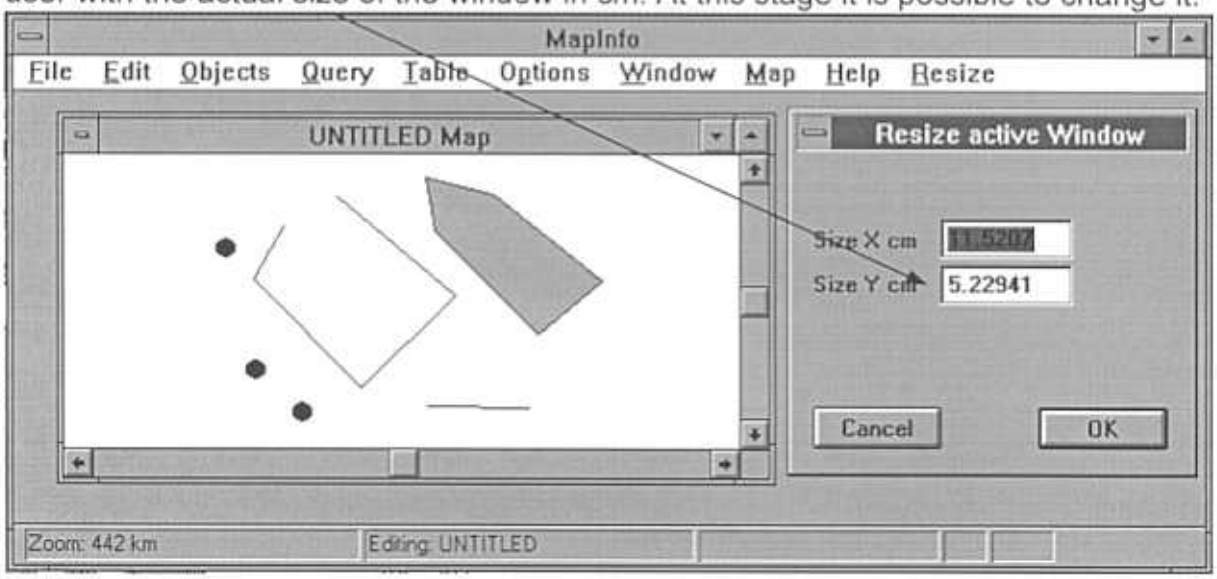
3.7.1 File Name
RESIZE1.MB

3.7.2 Purpose:
It re-sizes a MapInfo window to prepare for it to be included in a report or a presentation.

3.7.3 Origin:
Developped at SOPAC

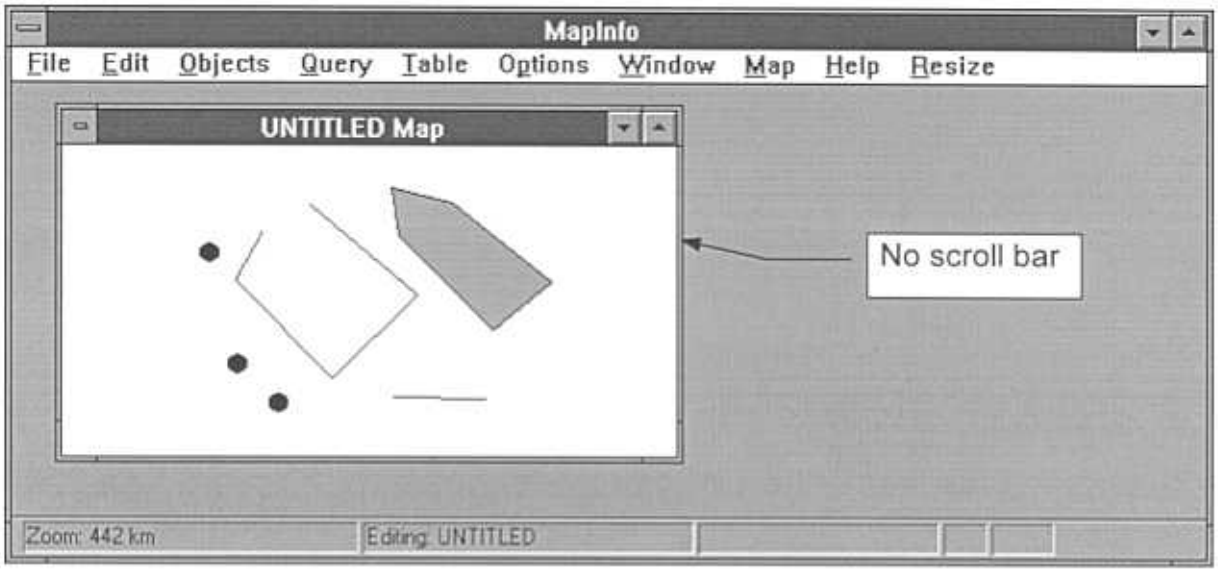
3.7.4 Additional files required
No Symbols necessary

3.7.5 Use
The utility must be launched when front window a Mapper. Then it will prompt the user with the actual size of the window in cm. At this stage it is possible to change it.



It keeps the content of the window identical at much as possible. When not possible, it will add information (extend the map) to keep the largest of the two dimensions.
It also removes the scroll bars so that the window will be filled and will not be changed during the export process.

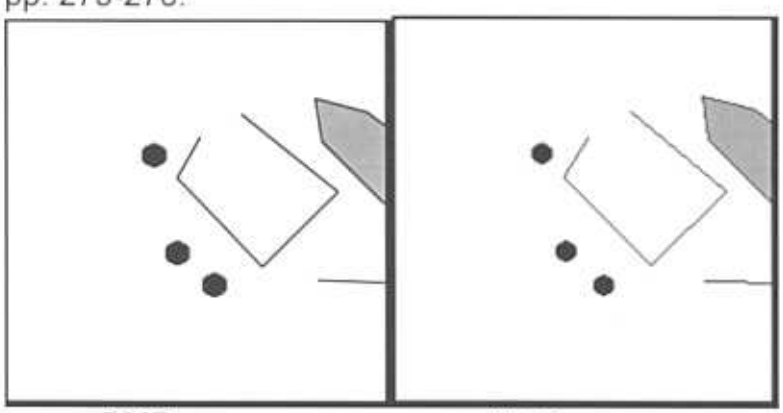
This utility performs a similar task as PortLand from KGM.



3.7.6 Known bugs, limitations

It is meant to be used in conjunction with the `Menu > File > Save Window As`.

MapInfo suggests two formats, one BitMap BMP (raster) the other Windows Meta File (vector). The two give similar results. For more information, see MapInfo Manual pp. 275-276.



BMP WMF

The advantage of this utility as opposed to the re-sizing while exporting the window is that it allows the user to control the content and the shape of the window.

3.8 TEST SYMBOLS

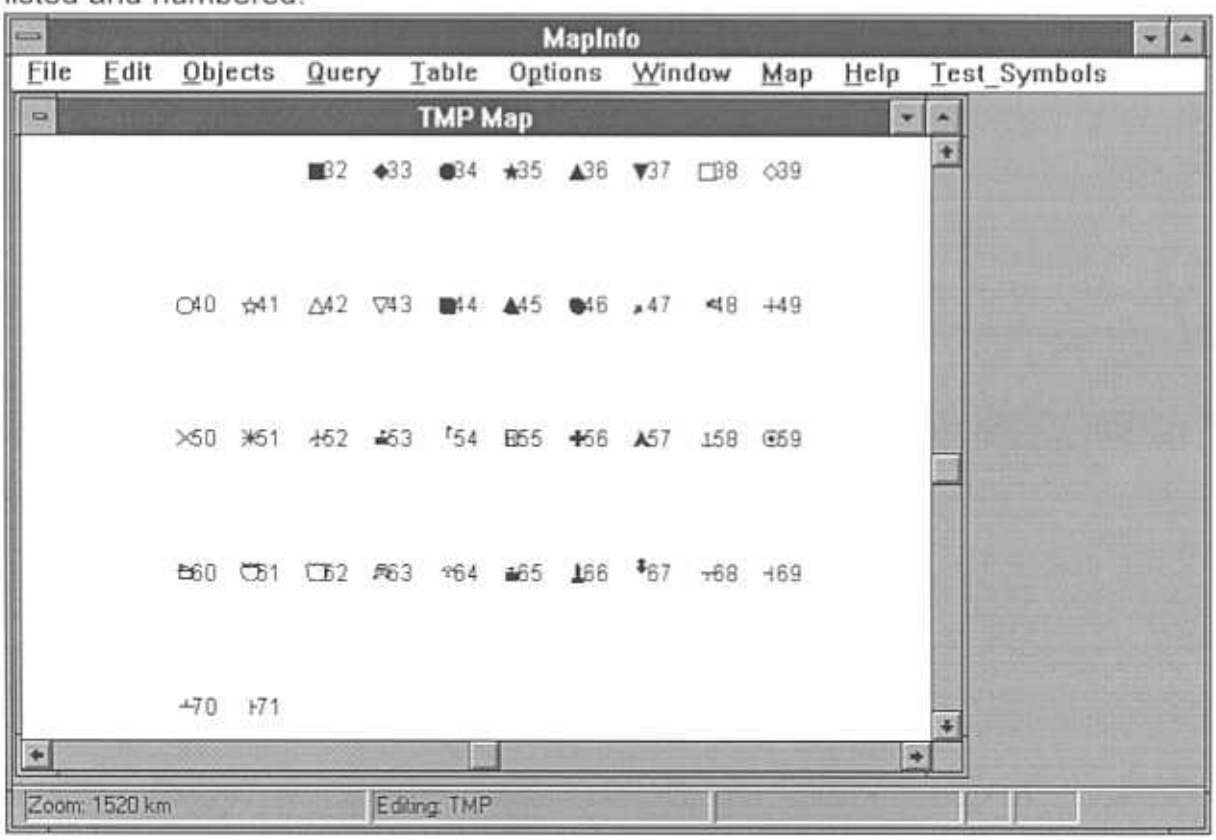
3.8.1 Filename
TST_SYMB.MB

3.8.2 Purpose:
MapInfo provides capability to edit, modify and change symbol sets. Symbols are the style of the Points put on a map. Several sets of Symbols can be stored, saved or changed. This utility displays the set of symbols actually active in MapInfo.

3.8.3 Origin:
Developped at SOPAC

3.8.4 Additional files required
It uses the active Symbols

3.8.5 Use
When launched, it produces and open a map called TMP with the active symbols listed and numbered.



3.8.6 Known bugs, limitations
There can be a conflict if a map named TMP is already opened.

3.9 VECTOR COMPONENTS

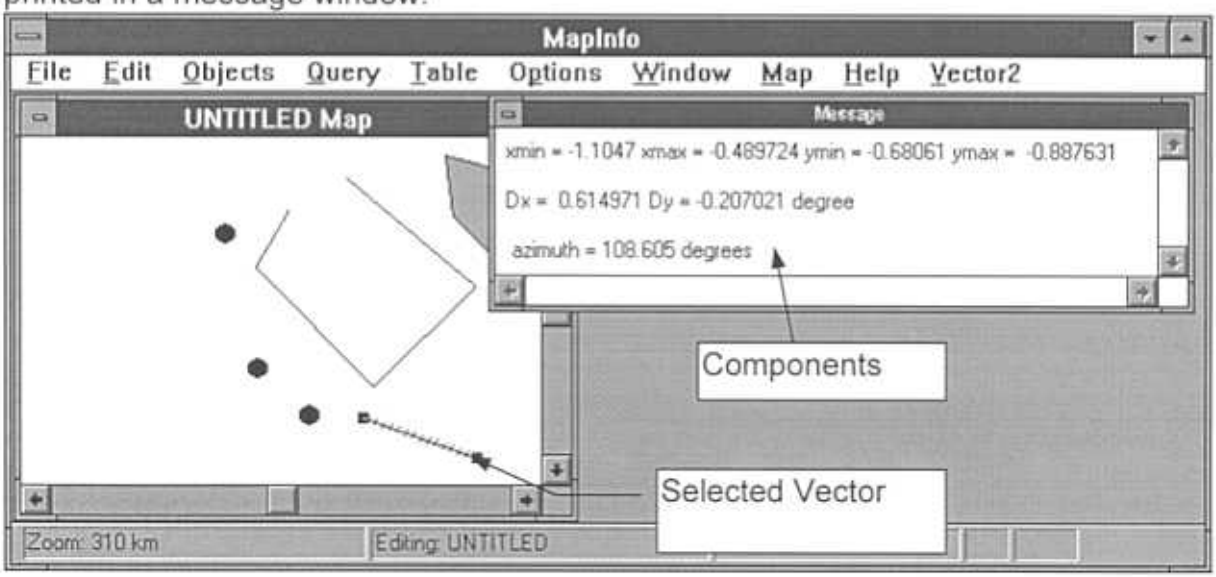
3.9.1 Filename
VECTOR2.MB

3.9.2 Purpose:
Measure the characteristics of a Line (a vector).

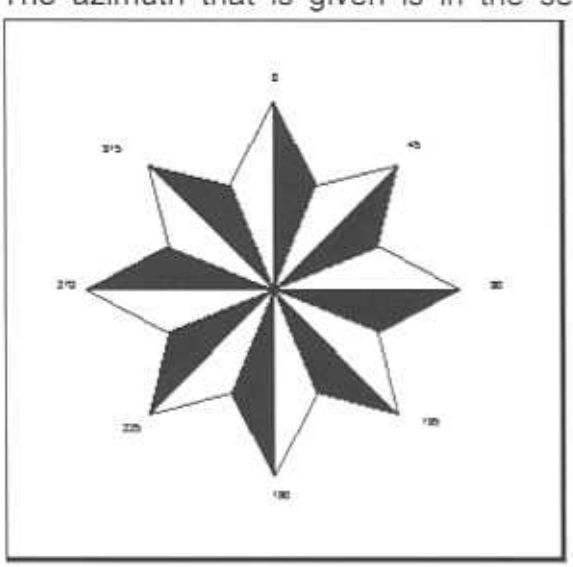
3.9.3 Origin:
Developped at SOPAC

3.9.4 Additional files required
No Symbols necessary

3.9.5 Use
Select a Line, or a series of lines. When done, the components of the vector are printed in a message window.



3.9.6 Known bugs, limitations
The azimuth that is given is in the sense of a wind direction or a ship heading. Therefore, North is 0, East is 90, South 180 and West 270.



It needs the file `ATAN2.MB` in the appropriate directory for instance `C:\MAPINFO`, to be re-compiled.

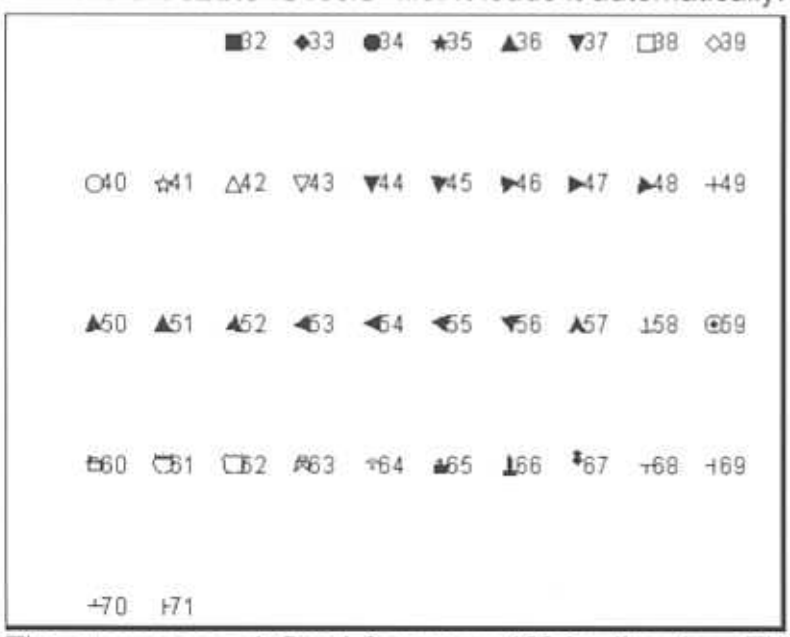
3.10 ADD ARROWS TO VECTORS

3.10.1 Filename
Vector3.MB

3.10.2 Purpose:
Add Arrows at the beginning, at the end or at both ends of a vector. If it is at the end, it indicates the direction of a movement, such as current.

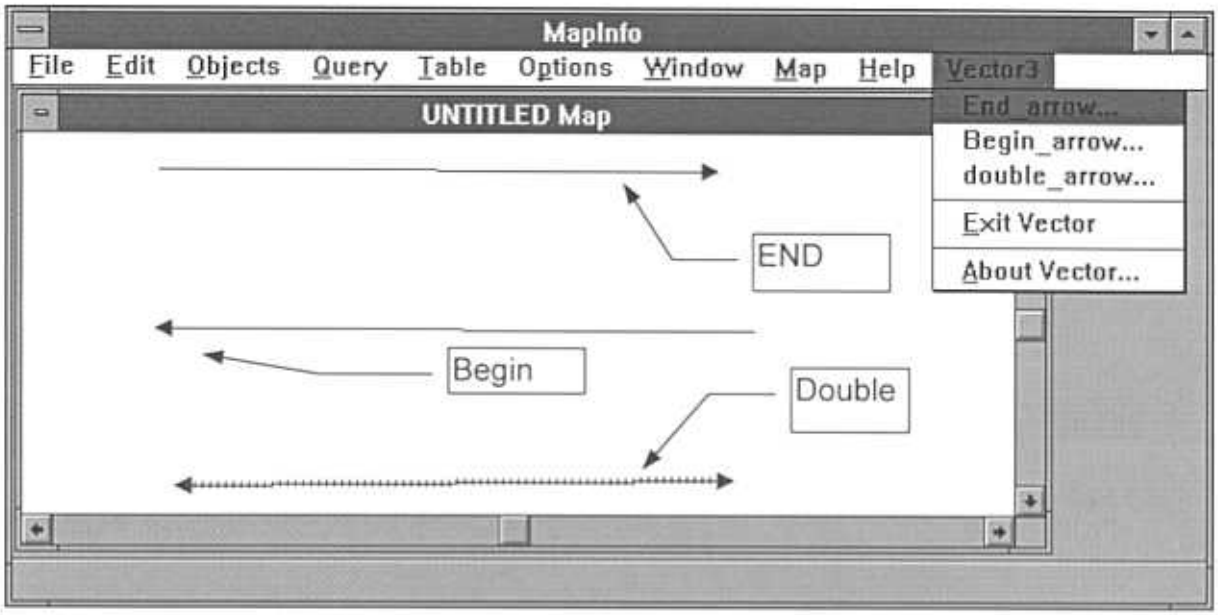
3.10.3 Origin:
Developped at SOPAC

3.10.4 Additional files required
It needs the `ARROWS.FNT` file. It loads it automatically.



The arrows are defined for every 10° in direction. This is a quick and dirty way of overcoming a limitation of MapInfo.

3.10.5 Use
The Menu proposes three possible actions: End, Begin and Double Arrows. Although it is not obvious in MapInfo, a line is a de facto vector, at it has a beginning and an end.



3.10.6 Known bugs, limitations

Depending on the way the utility ends, the Symbol file may not be updated at the end of the process. This can be tested with the TST_SYMB utility.

3.11 EXPORT NODES

3.11.1 Filename
EXP_PNTS.MB

3.11.2 Purpose:
It exports the coordinates of the nodes of Polygons, Polylines and points to a Text file. It also exports a third dimension taken from one column of the table.

3.11.3 Origin:
Developed at SOPAC

3.1 1.4 Additional files required
None

3.11.5 Use
The utility prompts the user to choose successively a TXT file to export to, a table to export from and a column of this table to be used as the third coordinate (Z).

3.11.6 Known bugs, limitations

3.11.7 Remarks
This utility is to be compared with EX_ASC from KGM.

3.1 1.8 Possible developments

3.12 MOVE BY A VECTOR

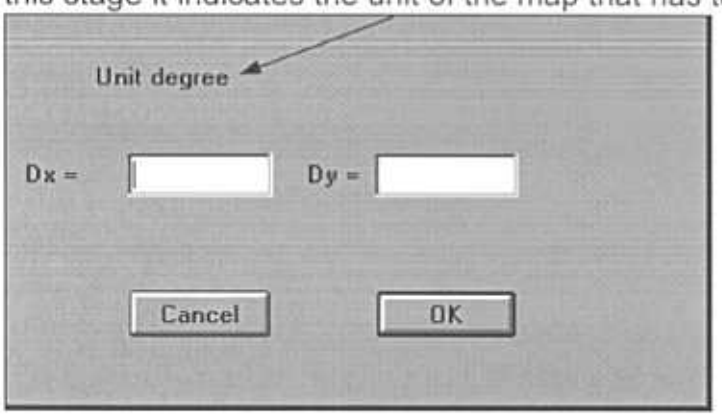
3.12.1 Filename
MV_VECT1.MB

3.12.2 Purpose:
Move all the objects contained in a Table by a vector defined by its components, Dx and Dy. The latter parameters can be measured using VECTOR2 utility.

3.12.3 Origin:
Developped at SOPAC

3.12.4 Additional files required
None

3.12.5 Use
The utility prompts for a table to be moved, then for the components of the vector. At this stage it indicates the unit of the map that has to be used.



3.12.6 Known bugs, limitations

3.12.7 Remarks

3.12.8 Possible developments

3.13 MOVE ONE TABLE BETWEEN 0 AND 360'

3.13.1 Filename
MV_WEST3

3.13.2 Purpose:
The utility moves all the objects contained in a table to fit in the 0-360" window.

3.13.3 Origin:
Developed at SOPAC

3.13.4 Additional files required
None

3.13.5 Use
The utility prompts the user to choose a table to move.

3.13.6 Known bugs, limitations

3.13.7 Remarks

3.13.8 Possible developments

3.14 MOVE SEVERAL TABLE BETWEEN 0 AND 360'

3.14.1 Filename
MV_WEST4. MB

3.14.2 Purpose:
Moves several tables listed in a Text file to be moved to fit between 0 and 360".

3.14.3 Origin:
Developed at SOPAC

3.14.4 Additional files required
None

3.14.5 Use
The utility prompts the user to provide a text file with the names of the files, including their complete path.

C:/TMP/TEMP

The extension TAB can be omitted.
All the tables will be closed and the utility opens the tables one by one. It gives a control on which table is being moved in the message window.

3.14.6 Known bugs, limitations
All the objects to be moved should not cross the 0" limit. They should then be cut before the work is done. This can be done by using the Menu>Object>Set Target then Menu>Object>Split command.
The tables are automatically packed before starting the work.

3.14.7 Remarks

3.14.8 Possible developments

3.15 BREAK REGION \:

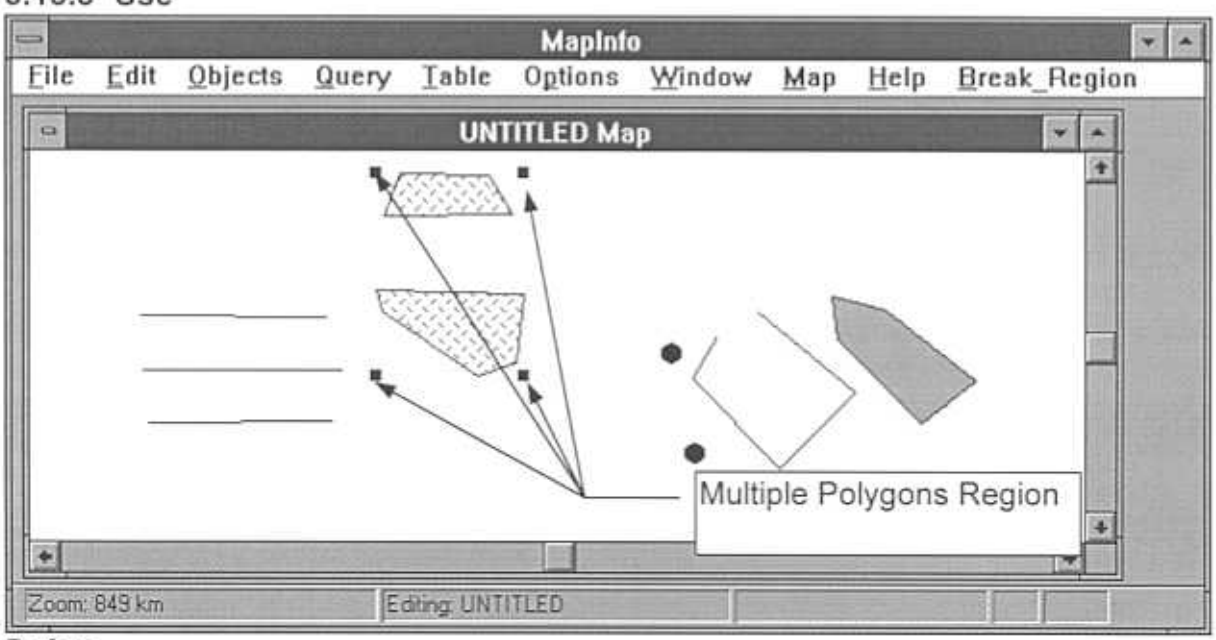
3.15.1 Filename
BRK_REG.MB

3.15.2 Purpose:
Polygons are handled in MapInfo as Region, which can contain several Polygons. Sometimes, it is necessary to have only single polygon regions. One way to do it is to be careful while creating the maps, but some functions of MapInfo generate multiple polygons regions, such as the Menu>Objects>Split. There is therefore a need to break them into individual polygons, or single polygon regions.

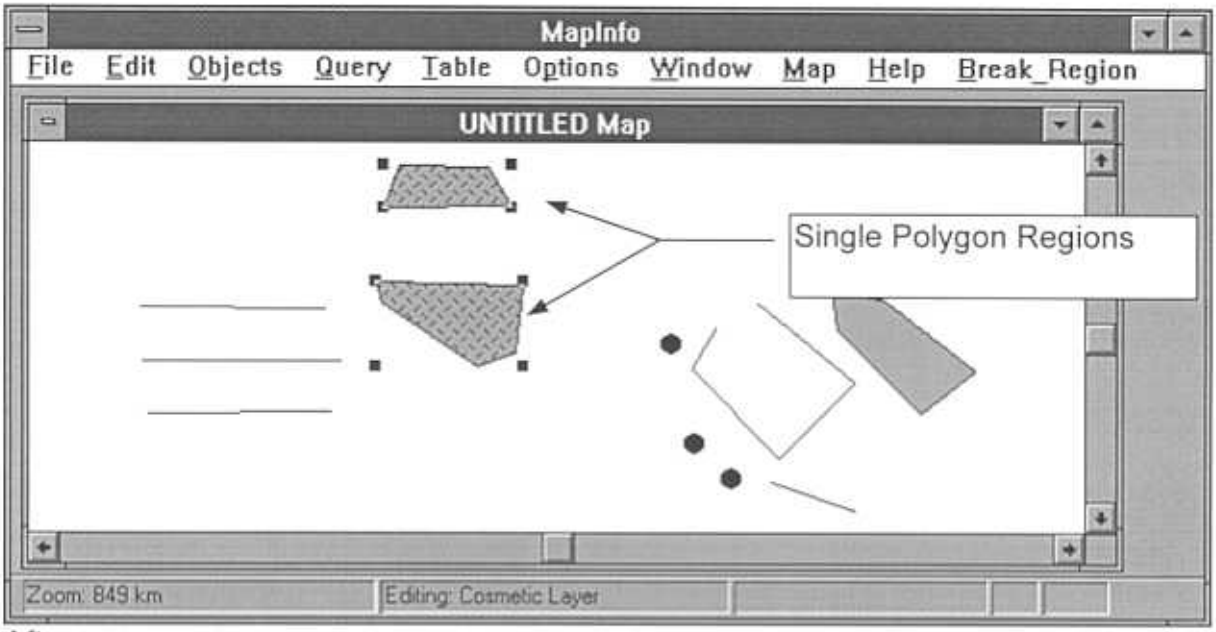
3.15.3 Origin:
Developped at SOPAC

3.15.4 Additional files required
None

3.15.5 Use



Before



After

3.15.6 Known bugs, limitations

3.15.7 Remarks

3.15.8 Possible developments

3.16 CLEAN POLYGONS

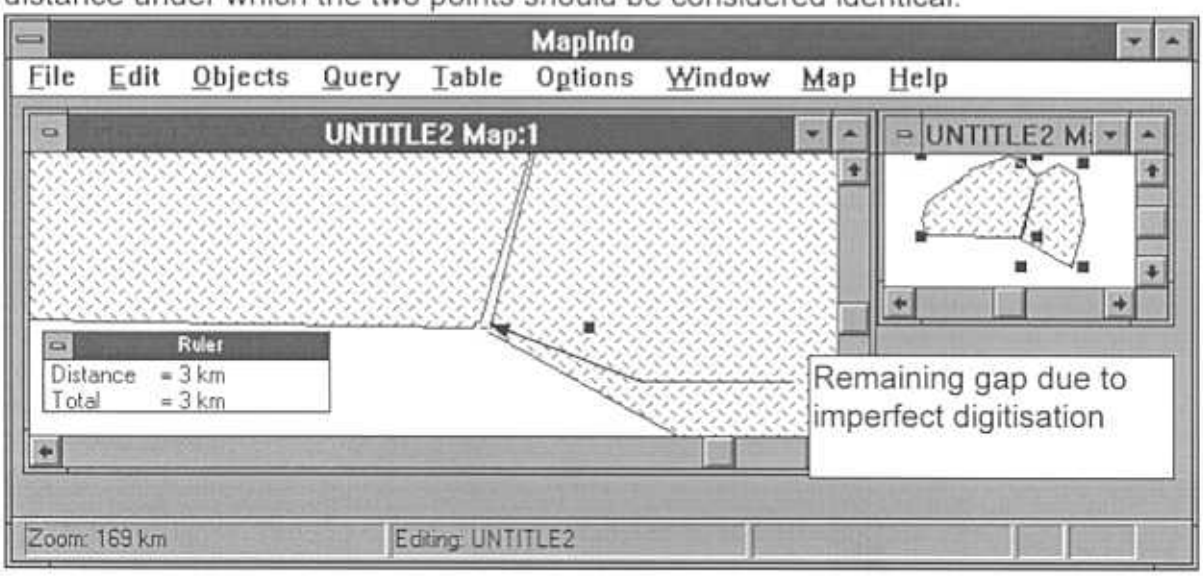
3.16.1 Filename
CLEAN1.MB

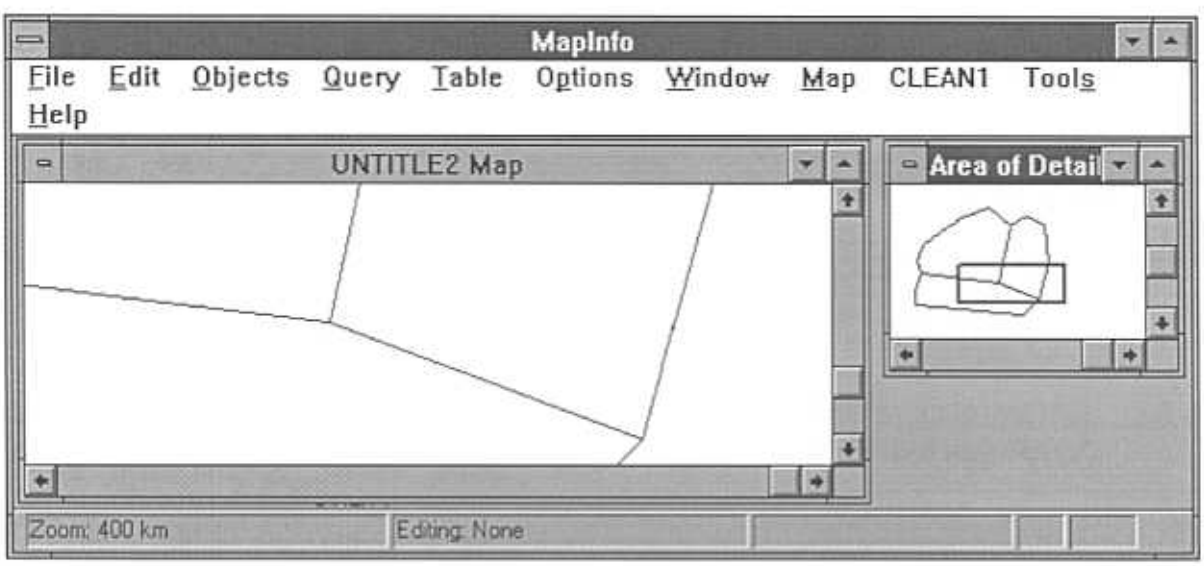
3.16.2 Purpose:
To clean adjacent polygons by putting the nodes at the same exact location.

3.16.3 Origin:
Developped at SOPAC

3.16.4 Additional files required
None

3.16.5 Use
The user has to select two adjacent polygons. The utility prompts for a minimum distance under which the two points should be considered identical.





3.16.6 Known bugs, limitations

It works actually only on two polygons.

3.16.7 Remarks

3.16.8 Possible developments

Extension to an unlimited number of polygons.

IDENTICAL WINDOWS

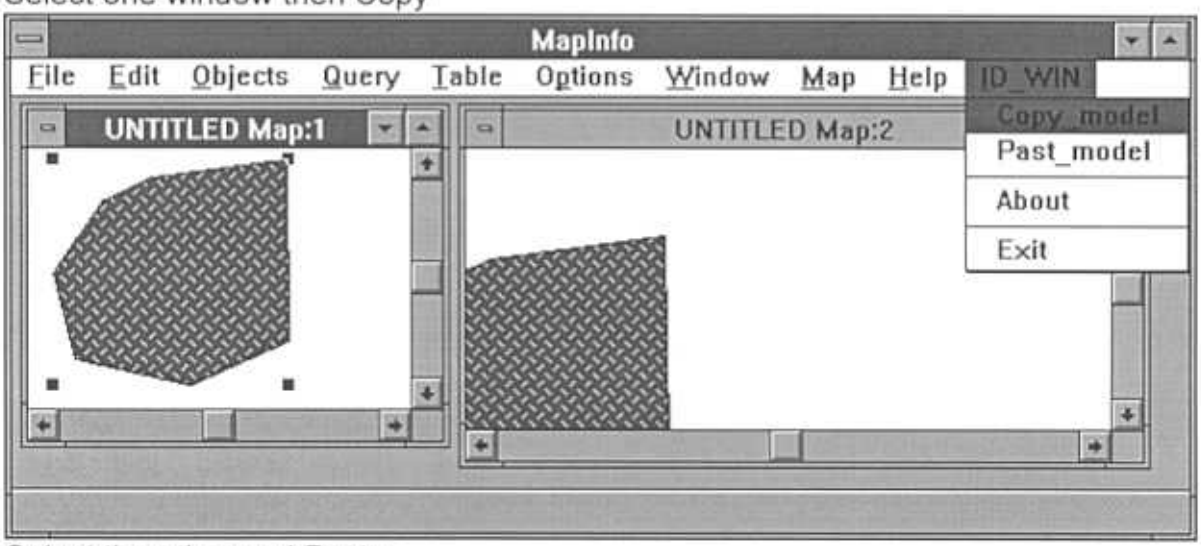
3.16.9 Filename
ID_WIN

3.16.10 Purpose:
To copy the geographical and geometrical characteristics of a window and paste them to another one.

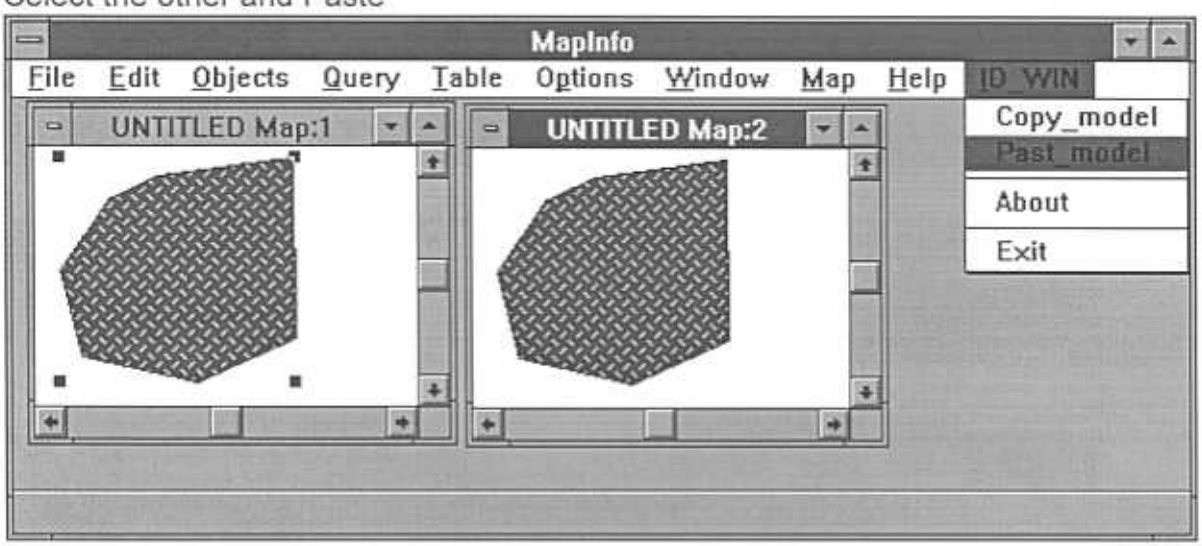
3.16.11 Origin:
Developped at SOPAC

3.16.12 Additional files required

3.16.13 Use
Same syntax as Copy and Paste.
Select one window then Copy



Select the other and Paste



3.16.14 Known bugs, limitations
It might not work for non Earth documents.

3.17 IMPORT MULTIPLE MIF

3.17.1 Filename
IMPORT1.MB

3.17.2 Purpose:
It allows to generate a text file with the names of the MIF files to import, then it imports successively all the MIF files and saves them as MapInfo standard format.

3.17.3 Origin:
Developed at SOPAC

3.17.4 Additional files required
A text file that contains the name and location of the MIF Files to import.

3.17.5 Use
When launched, this utility opens a DOS window. This is to let the user select the files to be imported.

1. GO INTO THE PROPER DIRECTORY
2. TYPE 'DIR *.MIF /B > MYFILE.TXT
3. TYPE EXIT TO RETURN TO MAP INFO

Then the utility prompts the user to open the file, select the one just created.

3.17.6 Known bugs, limitations
The progress bar requires to have MapInfo in the forefront.

3.17.7 Remarks
Instead of the command DIR *.MIF /B, any other command can be used to generate the text file. For instance in the case where the all files are not in one single directory (DIR/S/B). In any case, it is possible to edit the file before opening it in MapInfo, to add or remove some of the files not to be imported. The DOS part of the utility is just an example of one way to create a file list.

3.17.8 Possible developments

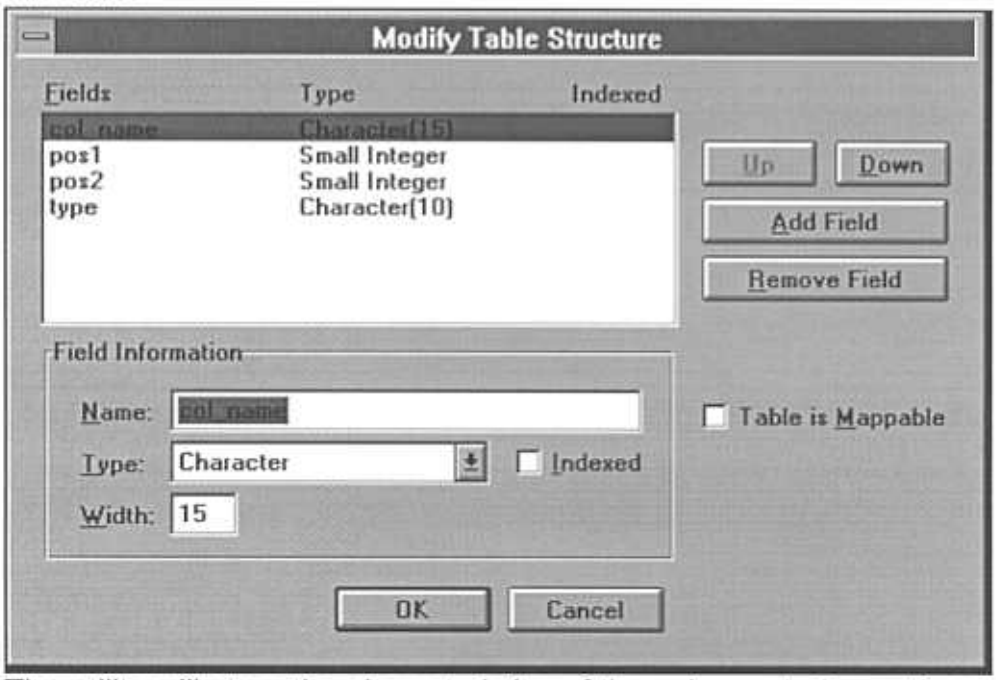
3.18 FIXED LENGTH ASCII IMPORT

3.18.1 FILE NAME:
fixed_im.mbx

3.18.2 DESCRIPTION:
This App will import into a fixed length ASCII file into a MapInfo table.

3.18.3 ORIGIN:
KGM

3.18.4 USE:
There needs to be a temporary empty table named `TAB_DEF` open with the following structure:



The utility will store the characteristics of the columns to import in `TAB_DEF`. If there is already one available, the user is prompted to open it. If none are already developed, a standard one is in the utility directory. The table will be emptied at the end of the use of the utility (when Menu>Fixed Length Import>Exit Application).
As a result it will create a new table called `NEW_TABLE`. The user will have to save it.
Only the values are imported. There is no way to import decimal number so far. In the case of decimal values, they have to be entered either as float or as integer. If they come from a FORTRAN generated file, some modification will have to be done later.
The resulting table is not mappable at this stage. If some columns contain latitudes and longitudes, the user will have to

1. Make the table Mappable (`Menu Table>Table Maintenance>Table Structure`)
2. Create points (`Menu Table>Create Points`)

3.19 INTERSECTION OF 2 REGIONS

3.19.1 FILE NAME
int_reg.mbx

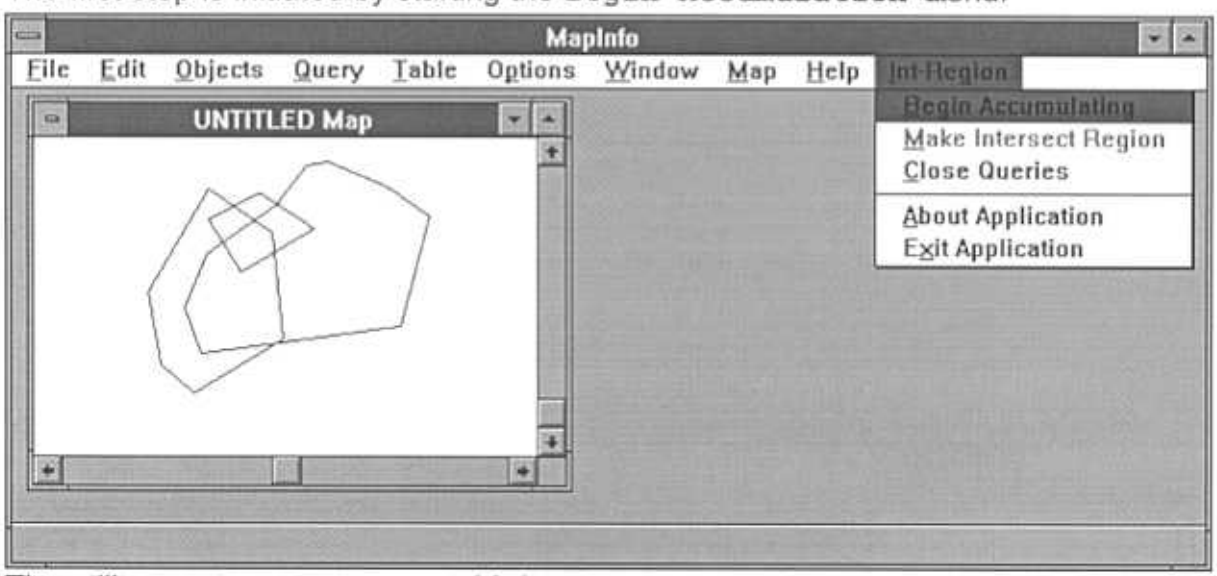
3.19.2 DESCRIPTION
This App will draw a region showing the intersection of two selected regions.

3.19.3 ORIGIN:
KGM

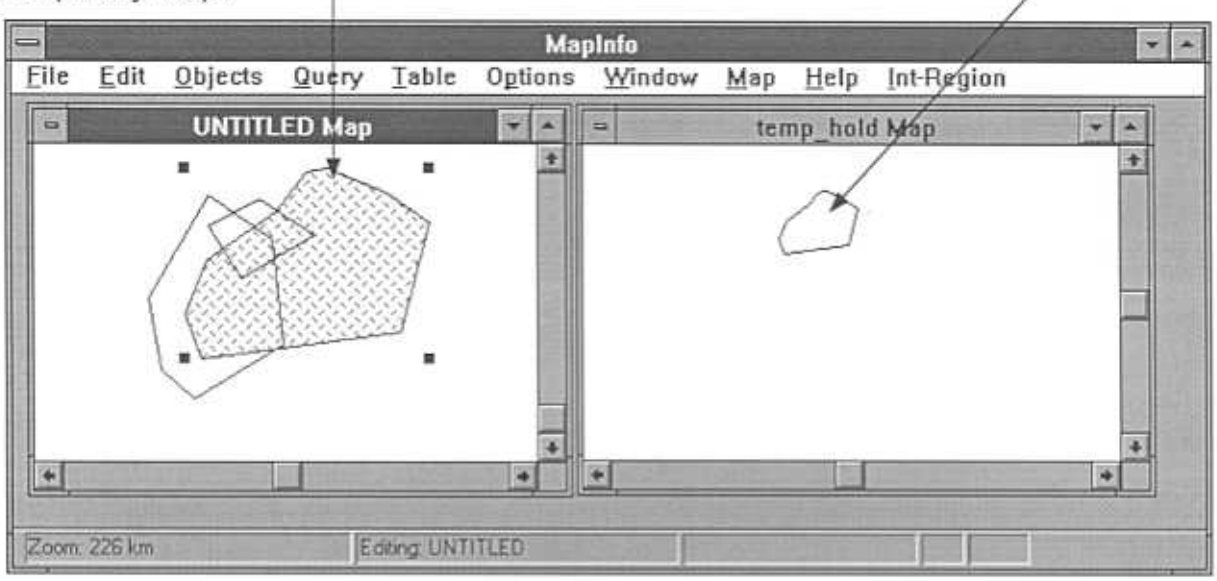
3.19.4 USE
Once launched, the utility creates a menu. The intersection is done in two steps.

1. Accumulation of the region
2. Intersection of the selected regions

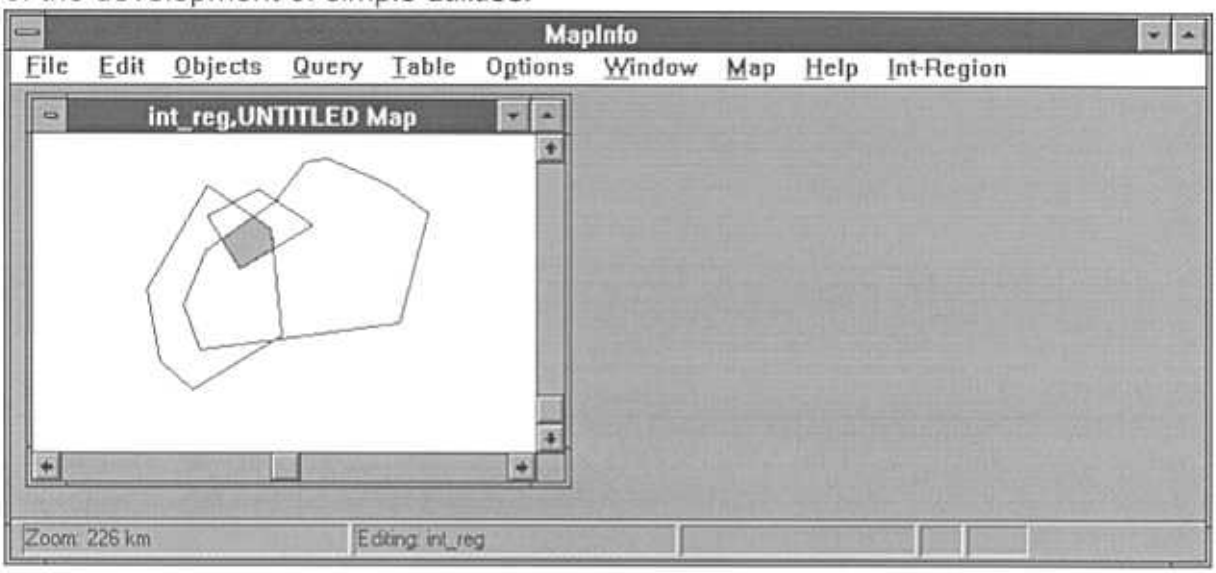
The first step is initiated by starting the `Begin Accumulation` menu.



The utility creates a temporary table/map
By selecting one object from the original map it is automatically copied in the temporary map..



The second by making the Menu `Make Intersection` active.
The result is put in a table called int_reg.
This utility shows how MapInfo can go toward a more comprehensive GIS at the cost of the development of simple utilities.



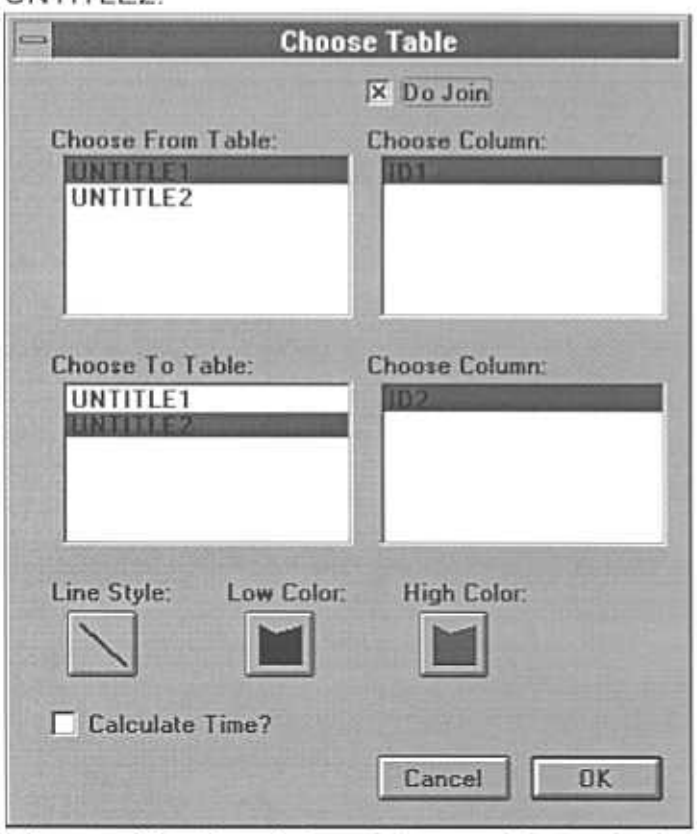
3.20 DRAW LINES TO POINT

3.20.1 FILE NAME:
Instopnt.mbx

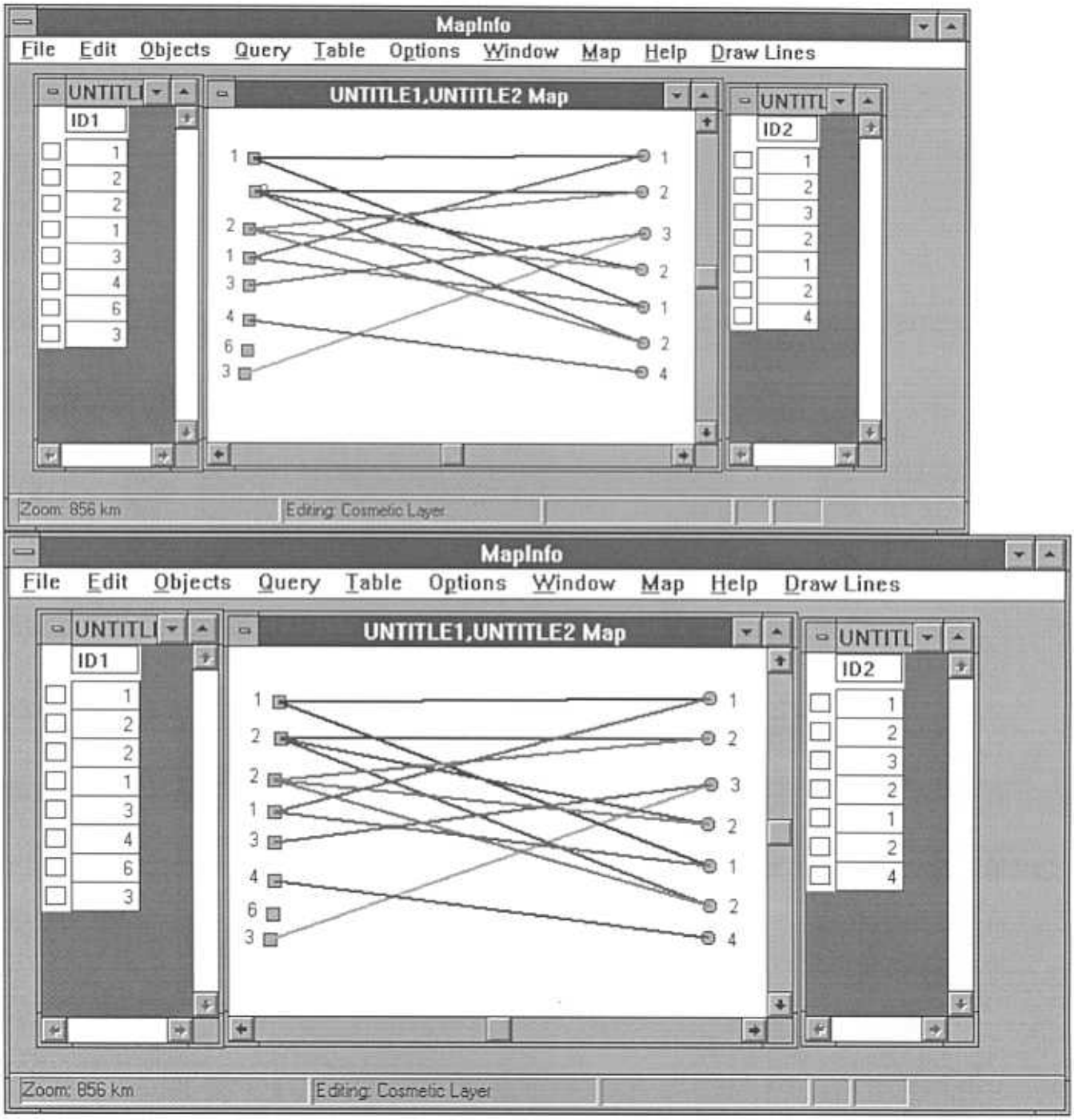
3.20.2 DESCRIPTION:
This App will draw lines from a central point out to a table with objects within it.

3.20.3 ORIGIN:
KGM

3.20.4 USE:
The utility will connect points from two tables with the same value in respective columns. In our example below, ID1 from UNTITLE1 will be matched with ID2 from UNTITLE2.



The result is show below. The result is put in the cosmetic layer. The utility first clears the cosmetic layer so any object such as autolabels will be erased. As well, the result will have to be stored before the utility is run again.


MapInfo
File Edit Objects Query Table Options Window Map Help Draw Lines
UNTITLI
ID1
1
2
2
1
3
4
6
3
UNTITLE1,UNTITLE2 Map
1 1
2 2
2 3
1 2
3 1
4 2
6 4
3
UNTITL
ID2
1
2
3
2
1
2
4
Zoom: 856 km
Editing: Cosmetic Layer
MapInfo
File Edit Objects Query Table Options Window Map Help Draw Lines
UNTITLI
ID1
1
2
2
1
3
4
6
3
UNTITLE1,UNTITLE2 Map
1 1
2 2
2 3
1 2
3 1
4 2
6 4
3
UNTITL
ID2
1
2
3
2
1
2
4
Zoom: 856 km
Editing: Cosmetic Layer

3.21 CONNECT POINTS

3.21.1 FILE NAME:
conndots.mbx

3.21.2 DESCRIPTION:
This App will connect points from a tableusing either objects or lat/long columns.

3.21.3 ORIGIN:
KGM

3.21.4 USE:
It does about the same thing as Joint_pnts. The main difference is that instead of working on a selection, it works on entire table. Also it has two options, either it joints the points defines as objects, or tit makes a line based on latitudes and longitudes of the points even if they are not on the map as object. The table must be mappable to work.

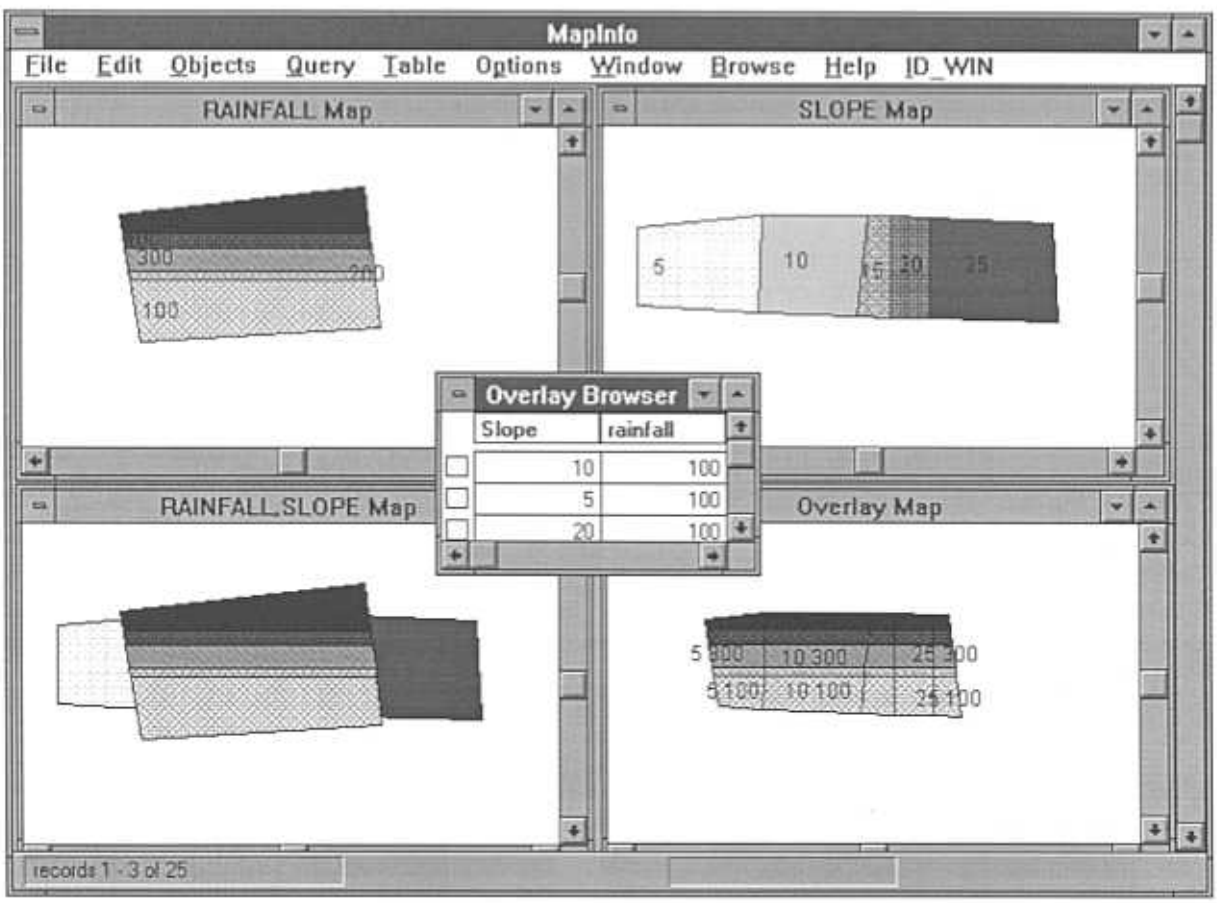
3.22 OVERLAY TABLES

3.22.1 FILE NAME:
overlay.mbx

3.22.2 DESCRIPTION:
This App will create an overlay of two tables.

3.22.3 ORIGIN:
KGM

3.22.4 USE:
This is a typical GIS application. Given two tables, in the example rainfalls and slopes. Slopes with values 100,200,300,400,500 from left to right and slopes with values 5, 10,15,20 and 25, from bottom to top. The utility creates a new table called `OVERLAY` with all the columns of the original tables. It will then break the polygons in smaller ones, each one with the two parameters of the original tables.



3.22.5 Bugs, Limitations
Note that this program was not thoroughly tested so it is possible that problems are encountered with the program or MapInfo itself. A known problem is the fact that MapInfo is allocating memory without freeing it up again, causing loss of memory every time you run it. The only way to free this memory is exiting MapInfo completely.

If you find out which statement (maybe the overlap function?) is causing this let me know will you ? Another problem you might stumble on is that the tables you are overlaying may not contain deleted records, pack them first!