

MODIFICATIONS TO THE MAPBASIC LANGUAGE

introduced with versions 4.0 to 6.5

Compilation of the MB help files by

Jacques Paris

02/2002

Overview of New Features in MapBasic 4.0

Animation Layers

The Add Map and Remove Map statements were enhanced to add support for a special map layer type known as the animation layer. If your application makes frequent changes to one of the layers in your map (for example, if you update point locations using signals received through a GPS receiver), you can speed up map redraws by designating an animation layer. For an example, see the sample program ANIMATOR.MB.

Array Sizes

When a MapBasic application runs on a 32-bit version of MapInfo, arrays can hold up to 32,767 items. Arrays are still limited to 7,000 elements under Windows 3.1.

ButtonPad (Toolbar) Improvements

The MapInfo 4.0 user interface now calls ButtonPads "toolbars" (e.g. the Options > Toolbars command). The terms "toolbar" and "ButtonPad" mean exactly the same thing. However, the MapBasic language syntax still uses ButtonPad as a keyword; in other words, the Create ButtonPad and Alter ButtonPad statements have not been renamed.

MapInfo 4.0 allows the user to "dock" toolbars to the top of the screen. To dock a toolbar through MapBasic, use an Alter ButtonPad... Fixed statement.

Each button can have a ToolTip (a small help message that appears next to the cursor when you leave the cursor positioned over a button). ToolTips are defined through the HelpMsg clause in the Create ButtonPad and Alter ButtonPad statements. To define a ToolTip, add \n to the end of the status bar help message, followed by the ToolTip text. For example:

```
HelpMsg "This button generates a report\nGenerate Report"
```

A new function, ButtonPadInfo(), returns information about ButtonPads.

Custom ToolButtons now support two new drawing modes: DM_CUSTOM_POLYGON draw-mode allows the user to draw a polygon, and DM_CUSTOM_POLYLINE allows the user to draw a polyline. For details, see Alter ButtonPad.

The Run Menu Command statement now has an optional ID clause, which you can use to make a custom toolbutton the active tool. (In version 3.0, you could use this statement to select a standard button, but not a custom button.)

Three new functions, SearchPoint(), SearchRect(), and SearchInfo(), search for map objects at a specific x/y map location or a rectangular map region. These functions are particularly useful when used with a custom ToolButton: After you let your user click on the map using a custom ToolButton, use one of the new Search functions to determine what map object(s) the user clicked on.

MapInfo now provides a wide assortment of ButtonPad icons. Most of the new icons do not appear in the standard user-interface at all; these icon resources were added to the product so that MapBasic developers would be able to choose from a wide variety of icons without having to manipulate DLL

files. ID numbers for all of the new icons are listed in ICONS.DEF. To see a demonstration of the new icons, run the sample program Icon Sampler (ICONDEMO.MBX).

Date and Number Formatting

In earlier versions of MapInfo, dates and numbers were always formatted according to US conventions. (Dates were always displayed as Month/Day/Year, and numbers were always formatted with the period as the decimal separator.) MapInfo 4.0 displays dates and numbers according to the formatting options set up on the user's computer. This enhancement is particularly useful to users outside of the US.

The functions `Format$()`, `Str$()`, and `StringToDate()` have been enhanced as part of the support for non-US formatting. For example, if you pass a date value to the `Str$()` function, the result string follows whatever date-formatting options are set up on the user's computer.

A new statement, `Set Format`, allows you to force MapInfo to adhere to use US formatting conventions, regardless of how the user's computer is set up.

A new function, `NumberToDate()`, returns a date value given an eight-character Integer date expression (such as 19951231).

DDE Server Improvements

When MapInfo acts as the server in a DDE conversation, the client can perform a peek request to query the value of a MapBasic expression. For details, see the MapBasic User's Guide, Chapter 11.

A new handler, `RemoteQueryHandler()`, gives MapBasic applications more flexibility in handling peek requests. This applies in situations where the MapBasic application is the server in a DDE conversation.

Dialog Boxes

It is now easier to store and retrieve values in custom dialog boxes. In earlier versions, you could only store text strings in `EditText` controls; this restriction forced you to convert numbers to strings before issuing the `Dialog` statement, and forced you to convert strings back to numbers after the `Dialog` statement. Beginning with version 4.0, the `Dialog` statement can work directly with numeric values. If you want an `EditText` control to display a number, specify a numeric variable in the `EditText` control's `Value` clause and `Into` clause.

A new statement, `Alter MapInfoDialog`, allows you to disable or change the value of controls in standard MapInfo dialog boxes.

When you include `StaticText` controls in a custom dialog, you can specify a `Width` clause and a `Height` clause. Specifying a height of 16 or larger allows text to wrap onto multiple lines.

Digitizer Improvements

In earlier versions of MapInfo, only MapInfo's standard drawing tools could be used with a digitizing tablet. In version 4.0, you can use any tool with a digitizing tablet, even custom MapBasic ToolButton tools. Also, the Set Digitizer statement has been enhanced, so that a MapBasic program can turn Digitizer Mode on or off.

DXF Import/Export Improvements

The Import and Export statements have been enhanced to provide better support for DXF files. MapBasic programs can now specify all aspects of an import/export operation programmatically. (In earlier versions, some aspects of a DXF import/export operation would require the user to fill in a dialog box.) Also, AutoCAD version 13 DXF files are now supported.

Integrated Mapping

You can easily integrate MapInfo windows into applications written using other languages, such as Visual Basic, PowerBuilder, or C. For details, see [Integrated Mapping](#).

Map Window Improvements

MapInfo 4.0 provides greatly-enhanced map labeling. Enhancements to the Set Map statement allow you to control labeling through MapBasic.

The Set Map statement includes several new clauses, including a Clipping clause that lets you specify a clipping object.

Text objects on Map and Layout windows can appear with new text styles, including Halo, Shadow, All Caps and Expanded. For details on the new text style options, see [Font](#) clause.

MapInfo 4.0 supports transparent fill patterns (patterns that have a transparent background). Transparent fill patterns make it easier to overlay region layers without obscuring the lower layers. To specify a transparent fill style, use -1 as the background color when calling the MakeBrush() function, or omit the background color argument when using the Brush clause.

MapInfo 4.0 supports three types of symbols: MapInfo 3.x symbols, TrueType font symbols, and custom bitmap symbols. To support the new symbol types, the Symbol clause has been expanded and there are two new functions: MakeFontSymbol() and MakeCustomSymbol().

When you use the Close Window statement to close a Map window, you now can include an optional Interactive clause. Use the Interactive clause if you want to prompt the user to save themes or cosmetic objects. This enhancement also applies to these statements: Close Table, Alter Table, Pack Table.

The MapperInfo() function can now determine the number of thematic layers in the window. Also, MapperInfo() can return information about the window's CoordSys.

You can now duplicate (or "clone") a Map window. For an example, see [Run Command](#).

The Save Window statement now allows you to set the image size explicitly.

When a user clicks a map object to drag it, MapInfo makes the user wait; this delay prevents users from dragging objects accidentally. The Set Drag Threshold statement sets the duration of the delay.

The Set Map statement now allows you to reposition a map without causing the entire window to redraw. If you include the optional Smart Redraw clause, the map window behaves as if the user had repositioned the map using the Grabber tool or the window scroll-bars. MapInfo redraws only the portion of the window that needs to be redrawn.

Menu Improvements

MapInfo 4.0 for Windows supports shortcut menus (menus that appear if the user clicks the right mouse button). To modify a shortcut menu, use the same statements you would use to modify a conventional menu. For example, you can use the Alter Menu ... Add statement to add custom menu items to a shortcut menu.

Each shortcut menu has its own name and ID number; see Alter Menu for a list of the names and IDs.

MapInfo 4.0 allows you to create 96 menus (including all of the standard menus); earlier versions allowed only 48 menus.

MapInfo's File menu now contains a most-recently-used (MRU) list. The MapBasic Create Menu statement can use the control code "\$" to build the MRU list into a menu. Similarly, the Create Menu statement can use the control code ">" to build the list of open windows into a menu.

Two new functions, MenuItemInfoByID() and MenuItemInfoByHandler(), return information about menu items.

Remote Data Access (Formerly SQL DataLink)

With earlier versions of MapInfo, access to remote databases was only available through the SQL DataLink add-on package. MapInfo 4.0 provides built-in access to remote data, and the MapBasic language includes statements and functions specific to remote-data access. For an introduction to remote-data access, see Chapter 7 of the MapBasic User's Guide. For details on remote-data statements and functions, see the MapBasic Reference or online Help; most related statement and function names start with Server (e.g. theServer Begin Transaction statement).

Seamless Tables

MapInfo 4.0 supports a new class of tables, known as seamless tables. For an introduction to seamless tables, see the MapInfo documentation. The new GetSeamlessSheet() function lets you prompt the user to select one sheet from a seamless table. See also: Set Table and TableInfo().

Table Metadata

MapInfo 4.0 allows you to store metadata in a table. In other words, you can store information about the table in the .TAB file, instead of storing it in rows and columns. See the new Metadata statement and GetMetadata\$() function.

Miscellaneous New Features in MapBasic 4.0

The Beep statement sends a sound to the speaker.

The Create Frame statement has been enhanced. Now you can create a frame object that is not affected by the aspect ratio of the Map window it represents; this means that it is easier to make the map fill the frame.

The FileAttr\$() function now allows you to determine the file size, in bytes.

New string functions, FormatNumber\$() and DeformatNumber\$(), allow you to add or remove thousands separators (e.g. commas) from strings that represent numbers.

A new system event-handler, ForegroundTaskSwitchHandler, lets you detect when MapInfo gets or loses the focus (i.e. when MapInfo becomes the active application or another application becomes active).

The new NumAllWindows() function returns the count of all MapInfo windows, including ButtonPad windows.

Using special codes in MENU.DEF, you can use the Run Menu Command statement to display MapInfo's Preferences sub-dialogs without first displaying the main Preferences dialog. For example, the statement Run Menu Command M_EDIT_PREFERENCES_MAP displays the Map Window Preferences dialog box.

The SelChangedHandler procedure can now detect whether the user interrupted a selection by pressing Esc.

The Select statement's Into clause now supports an optional keyword NoSelect. If you include NoSelect, MapInfo performs the query without disrupting the current selection.

A new statement, Set Command Info, lets you store a value in memory; the next call to CommandInfo() will retrieve the value.

A new statement, Set Handler, lets you disable event-handler procedures, such as SelChangedHandler.

The Set Table and Set Window statements have various new clauses that allow you to restrict what the user can do with a table or a window. For example, you can set a window so that the user cannot close the window via the Control menu.

The Set Window...Help statement now gives you more control over the Help window.

The Set Window...Info statement lets you display a specific row of data in the Info window. You can also make the Info window read-only by including the optional ReadOnly keyword. For an example, see the MapBasic User's Guide, Chapter 6.

The SystemInfo() function can now return information about the operating environment (e.g. 16- vs. 32-bit Windows).

The `TableInfo()` function has been enhanced, allowing you to query several new table attributes. For example, specify `TAB_INFO_TABFILE` to determine the table's full directory path. You also can use `TableInfo()` to determine a table's `CoordSys`.

The `WinChangedHandler` procedure is now triggered whenever a map layer is added or removed.

`MapBasic` now lets you specify hexadecimal string constants, using the `&` prefix (e.g. `&H1A`). The `Val()` function now also supports hexadecimal notation.

The `&` operator also acts as a string-concatenation operator (similar to the `+` operator).

Overview of New Features in MapBasic 4.1

Microsoft Access Support

MapInfo 4.1 lets you work directly with Microsoft Access tables. For general information about this feature, see the MapInfo documentation.

Once you have opened an Access table in MapInfo, you use it as you would use any other table; therefore, most MapBasic statements do not have any new syntax to support Access. However, the Create Table, Commit Table, Open Table, and Register Table statements have been enhanced so that you can specify Access-specific arguments (password, etc.).

New Text and Labeling Features

MapInfo 4.1 provides four new functions that allow you to query map labels: LabelFindFirst(), LabelFindNext(), LabelFindByID() and LabelInfo().

The new CreateText() function allows you to create a text object using a single x/y coordinate pair (as opposed to the Create Text statement, which requires two x/y pairs).

Also, the Set Map statement has a new clause, PartialSegments, that corresponds to the new Label Partial Segments check box in MapInfo's Label Options dialog box.

The new sample MapBasic utility, LABELER.MBX, demonstrates some of the new labeling capabilities.

New statement: Reproject statement

The Reproject statement lets you specify the list of columns that should appear the next time a table is browsed. If you issue a Reproject statement, and then issue a Browse statement, the new Browser window will show only the columns that were listed in the Reproject statement.

New handler procedure: RemoteMapGenHandler

RemoteMapGenHandler is a special-purpose MapBasic procedure name, which is invoked through OLE Automation. If you are using OLE Automation to control MapInfo, and you call the MapGenHandler method, MapInfo calls the RemoteMapGenHandler procedures of any MapBasic applications that are running. The MapGenHandler method is part of the MapGen Automation model introduced in MapInfo 4.1.

Enhancements to the Save Window statement

The Save Window statement now allows you to generate new file formats (such as TIFF) and add copyright notices to the images.

Improvements to remote data syntax

A new statement, Server Set Map, allows you to change object styles.

The Server Link Table statement has also been enhanced.

OLE Automation enhancements

MapInfo's OLE Automation object model has been enhanced. The enhancements are primarily of interest to developers who are creating MapInfo ProServer applications; however, you can access the same object model using MapInfo Professional.

The MapInfo ProServer package includes more detailed documentation and program examples of creating ProServer applications.

Overview of New Features in MapBasic 4.1.2

Enhanced clause: Coordsys

Addition of the AFFINE clause to syntax1 and syntax2 of that clause

Syntax1

```
CoordSys Earth
  [ Projection      type,
    datum,
    unitname
    [ , origin_longitude ]
    [ , origin_latitude ]
    [ , standard_parallel_1 [ , standard_parallel_2 ] ]
    [ , azimuth ]
    [ , scale_factor ]
    [ , false_easting ]
    [ , false_northing ]
    [ , range ] ]
  [ Affine Units unitname, A, B, C, D, E, F ]
  [ Bounds ( minx, miny ) ( maxx, maxy ) ]
```

Syntax2

```
CoordSys Nonearth
  Units unitname
  [ Affine Units unitname, A, B, C, D, E, F ]
  Bounds ( minx, miny ) ( maxx, maxy )
```

A performs scaling or stretching along the X axis.

B performs rotation or skewing along the X axis.

C performs shifting along the X axis.

D performs scaling or stretching along the Y axis.

E performs rotation or skewing along the Y axis.

F performs shifting along the Y axis.

unitname is a string representing a distance unit of measure (e.g. "m" for meters); for a list of unit names, see Set Distance Units

Enhancements to the Set Window statement

The Set Window has a new setting, Smart Pan which changes the status of the windows panning.

When Smart Pan is turned on for a Map window or a Layout window, panning and scrolling use off screen bitmaps to reduce the number of white flashes. Smart Pan is off by default. When Smart Pan is on for a Layout window, it only affects the redraw when the Grabber tool is used.

When Smart Pan is on for a Map window there will be different effects depending on the method of moving the Map. The Grabber tool will automatically paint the exposed area as the user grabs and moves the map. The Map will move more slowly than when Smart Pan is off. A more complex Map will move more slowly. Scrollbars and Autoscrolling will perform similarly to the grabber tool, but the speed

of the scrolling is not affected by smart panning. When the MapBasic command Set Map is used to Center or Pan with Smart Redraw on, the Map window will change without white flashes unless the Map is repositioned so far as to require a complete redraw.

NOTE: If off screen bitmaps have been turned off, then Smart Pan in a Map window behaves like a Layout window.

New function: Time()

The Time function returns the current system time as a string in HH:MM:SS format. The time may be returned in 12 or 24 hour time format.

Overview of New Features in MapBasic 4.5

New date function : CurDate()

The Curdate() function returns a Date value representing the current date. The format will always be YYYYMMDD. To change the value to a string in the local system format use the FormatDate\$() or Srt\$() functions.

New pen pattern object style

The PenPattern () function returns an integer type variable containing the correct pattern number for interleaved or over lapped line styles. Use this function to get the proper line style for an already existing interleaved or overlapped line style.

New functio for setting a projection

The ChooseProjection\$() function displays the Choose Projection dialog and returns the selected coordinate system as a string. The returned string is in the same format as the CoordSys clause. Use this function if you wish to allow the user to set a projection within your application.

Overview of New Features in MapBasic 5.0

Continuous thematic shading, new statement: Create Grid

A grid surface theme is a continuous raster grid produced by an interpolation of point data. The Create Grid statement takes a data column from a table of points, and passes those points and their data values to an interpolator. The interpolator produces a raster grid file, which MapBasic displays as a raster table in a map window.

Comprehensive report writer, new statement: Create Report From Table

Seagate Crystal Reports: Even higher quality reports of tabular data, processed within MapInfo, can now be produced using this industry standard report writer. Crystal provides a highly intuitive environment for developing professional reports.

Cartographic legend, new statement: Create Cartographic Legend

The Create Cartographic Legend statement allows you to create and display cartographic style legends as well as theme legends for an active map window. Each cartographic and thematic styles legend will be connected to one, and only one, map window so that there can be more than one legend window open at a time.

new statement: Set Cartographic Legend

The Set Cartographic Legend statement allows you to set redraw functionality on or off, refresh, set the orientation to portrait or landscape, or change the frame order of an existing cartographic legend created with the Create Cartographic Legend statement. (To change the size, position or title of the legend window, use the Set Window statement.)

new statement: Alter Cartographic Frame

The Alter Cartographic Frame statement changes a frame(s) position, title, subtitle, border and style of an existing cartographic legend created with the Create Cartographic Legend statement. (To change the size, position or title of the legend window, use the Set Window statement.)

new statement: Add Cartographic Frame

The Add Cartographic Frame statement allows you to add cartographic frames to an existing cartographic legend created with the Create Cartographic Legend statement.

new statement: Remove Cartographic Frame

The Remove Cartographic Frame statement allows you to remove cartographic frames from an existing cartographic legend created with the Create Cartographic Legend statement.

New legend functions

LegendInfo()

Returns information about a legend.

LegendFrameInfo()

Returns information about a frame within a legend.

LegendStyleInfo()

Returns information about a style item within a legend frame.

Enhancements to existing statements

Live Access To Remote Databases: Register Table

You can now access data live from remote databases with the Register Table statement. When you specify the Type as ODBC, the Register Table statement tells MapInfo to examine the ODBC table and build a corresponding table file(filename.TAB).

Enhanced Statement for Grid Themes: SetMap

The Set Map has a new Inflect clause which overrides the inflection color:value pairs that are stored in the MIG file of a grid surface theme layer created by the Create Grid statement.

Enhanced Statement: SetWindow

The Set Window [ReadOnly | Default Access] statement has been extended to work with Browsers and Legends. The behavior is as follows. When Browsers are set to ReadOnly, the user can not edit data from within the Browser. When Legends are set to ReadOnly, the user can not double click on them and get the "Modify Thematic Map" or "Customize Graph Legend" dialogs. This works for the main legend and cartographic legends created with the Create Legend statement or Create Cartographic Legend.

Overview of New Features in MapBasic 5.5

New statement: SetDateWindow.

New function: DateWindow()

New Buffer function :CartesianBuffer()

New function: SphericalArea()

New function: SphericalDistance()

New function: SphericalObjectLen ()

New function: SphericalPerimeter ()

New function: CartesianDistance()

New function: CartesianArea ()

New function: CartesianObjectLen ()

New function: CartesianPerimeter ()

Improved statement: Set Window

Each window now has its own snap mode state and snap mode and tolerance is now settable for each mapper and layout window.

New statement: Relief Shade

New function: LegendFrameInfo()

Improved statement: Set Window

Control Printer from MapBasic

Improved function: WindowInfo()

Control Printer from MapBasic

New statement: CreateGrid

New function: Server_ConnectInfo()

See Appendix C in the MapBasic User's Guide for a discussion of enhancements to object processing.

Enhancements to existing commands

Alter Object statement

Add Cartographic Frame statement

Create Cartographic Legend statement

Create Grid statement

LayerInfo statement

LegendFrameInfo() function

MapperInfo() function

Server_Create Map statement

Server_Connect() function

Set Graph statement

Set Map statement

StringToDate() function

SystemInfo() function

WindowInfo()function

Year () function

MapBasic 6.0 provides the following improvements:

New for 3D Map

- Create Map3D statement
- Set Map3D statement
- Map3DInfo() function

Other New Commands

- Objects Check statement
- Objects Enclose statement
- ConvexHull() function

Enhancements to existing statements/functions

HotLinks Enhancements

- Query HotLink options using the LayerInfo() function
- Optional Open Table clause for HotLinks
- Set Map clause for HotLinks
- CommandInfo() attributes for HotLinks

MS Access Enhancements

- Optional Open Table clause to Restrict Multiple MS Access Users
- Type clause for Commit Table statement sets MS Access version
- Type clause for Create Table statement sets MS Access version

Other Enhancements

- Align clause for Set Cartographic Legend statement
- Update for Set Map statements for Clip Region
- Clip Region attribute for MapperInfo() function
- Geographic Operation for Create Object statement
- Inset/Offset Clause for the Find Using statement
- Page settings for the Run Menu Command
- Printing/Export Paper Size and Margin Enhancements for the Set Window statement
- Papersize, Margin Attributes and Advanced Options for the WindowInfo() function

Overview of New Features in MapBasic 6.5

Upgrading your MapBasic Applications to version 6.5

MapInfo Professional Data and Settings Management- Application Data Files

Application data (appdata) files are the non-executable, non-user data files that MapInfo Professional uses during execution. The following files and directories are considered appdata for version 6.5:

Filename	Description
mapinfo.prf	Preference file
mapinfo.wor	Default workspace
startup.wor	Startup workspace
mapinfo.clr	Color file
mapinfo.pen	Pen file
mapinfo.fnt	Symbol file
custymb	Custom symbol directory
thmtmpl	Theme template directory
graphsupport	Graph support directory

Traditionally these files have been kept in the Windows directory or the Program directory. The strategy for 6.5 is to install application data files in a "per user location" and look for them in other areas as well.

This strategy allows support for sharing application data files between MapInfo versions/products. By relocating mapinfo.prj a user can share one custom projection files between different versions of MapInfo Professional.

The following files remain in the Windows Program directory:

Filename	Description
mapinfo.abb	Abbreviation file
mapinfo.prj	Projection file
mapinfo.mnu	Menu file

Keep in Mind:

- The installer never asks the user where they want to place application data files.
- The installer always runs the same way, whether the user has MI Pro 6.0 installed or not.
- There is not an "upgrade" install for 6.5 (i.e. you cannot install 6.5 into the same directory as 6.0, the installer will error).
- Application developers can move or copy files where they want, but MI Pro 6.5 will search for them only in these locations and in this order:

appdata_dir, local_appdata_dir, pref_dir, program_dir

For more information on data settings see:

A Glossary for Upgrading Applications
Application Data Files and Directories
Default Preference Paths
Registry Changes
Installer Requirements

New Statements and functions

For Prism Maps

Create PrismMap statement
Set PrismMap statement
PrismMapInfo() function

Enhancements for Object processing:

Objects Disaggregate statement
Objects Snap statement
Objects Clean statement

Create new object types:

Create Multipoint statement
Create Collection statement

Set Object styles for a mapped table:

Server Create Style statement

File Location functions:

LocateFile\$() function
GetFolderPath\$() function

Enhancements to existing statements/functions

Added identifiers for per row symbology in the MapCatalog:

Server Create Map statement

Invert selections :

Run Menu Command statement

Translucency per layer is supported for raster and grid files: and display coordinates in Military Grid Reference System format

Set Map statement

Display coordinates in Military Grid Reference System format

MapperInfo() function

The following functions and statements now support the new multipoint and collection object types:

Create Cartographic Legend statement

ObjectInfo() function
ObjectGeography() function
ObjectNodeX() function
ObjectNodeY() function
Objects Enclose statement
Combine() function
Alter Object statement

Enhancements for Object processing:

Gap detection for the Objects Check statement

New formats for saving windows:

New Formats for Save Window statement

Raster image support for registered tables:

Raster Image support for Register table statement