

## Use of the MindMap Menu bar

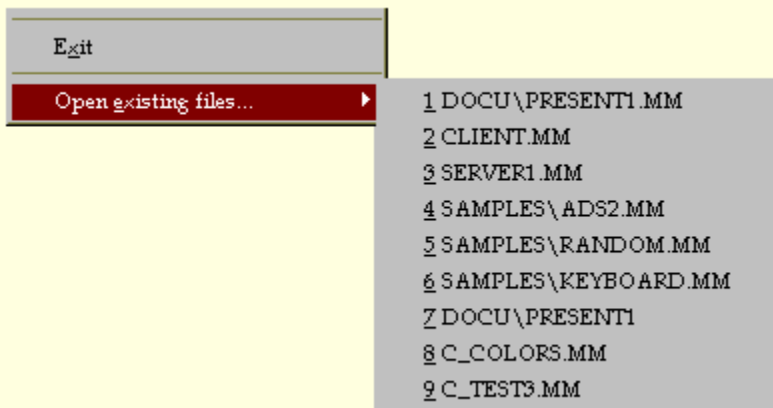
Before we begin to explain the MindMap menu bar, let us explain the essential commands and functions in a MS-Windows menu. MS-Windows is a graphical user interface (*GUI*) that can be used comfortably with a mouse. Nevertheless, you still have the option of using the keyboard of your computer for many functions. For example, when you are writing, it might be quicker to use a key combination for accessing the appropriate command. After starting MindMap, you can access the MindMap menu commands by pressing the ALT key and the underlined letter. E.g., *ALT+F* opens the menu of **F**ile. Cursor left or cursor right opens the menu of the immediately adjacent item.



Pressing one of the underlined letters in the menu takes you directly to the next command line; e.g., *ALT+F+O* shows the [File | Open](#) dialog box. The three dots behind a command inform you that there is a branch to a dialog box where the necessary actions can be done. For example, *ALT+F+O* shows a list of files that can be opened.

Some of the commands in the menus have a key combination at the end of the line. These shortcut keys branch to the command directly; e.g., *CTRL+N* shows the [New File](#) dialog box.

The triangle at the end of the line of a menu shows that there is a fly-out menu that will give you an additional selection for the chosen command. For example, the command **Open existing files** gives a list of MindMap files you have already saved. (This list of existing files is stored in the MINDMAP.INI file.)



The command **Open existing files** is only available if you have previously saved a MindMap application.

A grayed-out font in a menu indicates that this command is not available at the moment. For example, the commands **Cut**, **Paste** and **Delete** have a gray font as long as you have not selected a component. Selecting a component turns the gray font into a black one and gives you the option to cut, paste or delete the component.

The sign at the beginning of a command line acts as a toggle and shows that a special attribute is either on or off. Clicking on a command, e.g., [View | Toolbox](#) in the menu **View**, shows or hides the toolbox, so the sign is on or off.

You can close a menu selection by pressing the *Esc* key.

Please note, as in the example above, that some commands are only available when a component other than a page is selected.

## File | New

This command opens a dialog box with the choice of either **Application** or **Printer layout**.

By choosing **Application**, you can open a new application. The name of the actual application is shown at the top of your window in the caption bar. It should be noted that the default name (*as assigned by MindMap, Application 1*) will be used here until you have first saved the application, thereby defining a real name for the application.

The Printer layout allows you to define the appearance of the documentation you can produce for your application. MindMap is shipped with a collection of layouts, which can be used as-is or as templates for your own personalized layouts.

The standard printer layouts are stored in the file STANDARD.MMP.

MindMap is shipped with various standard printer layouts that are used if you print documentation of your application. The standard printer layouts are stored in the file STANDARD.MMP. You can change any layout or create your own printer layouts.

You can open multiple MindMap applications at the same time. The **Windows** menu shows the different MindMap applications you have open and allows you to switch between them.

### **Note:**

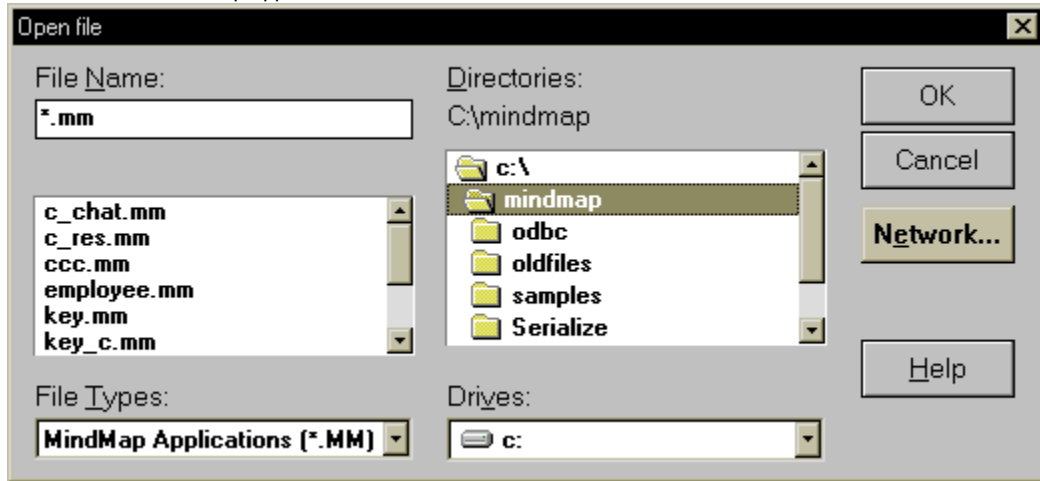
*Although you can have multiple MindMap applications open on your system, you can not have more than one instance of MindMap, itself, running at one time.*

If you click on the size symbol of your MindMap applications, you can size and arrange them on your screen. You can also use the **Window | Tile** or **Window | Cascade** commands.

## File | Open

This command opens a dialog box that permits you to open an existing MindMap application. MindMap shows a list of files in its working directory, normally "C:\MINDMAP". When you save a MindMap application to another directory or sub-directory, the working directory in MINDMAP.INI will be adjusted. This means that MindMap always tries to load an application from that directory in which you most recently saved a MindMap file. Note that MindMap only remembers the directory of the last file saved.

A list offers a choice of different **File types**: MindMap application files (\*.MM), Printer Layouts (\*.MMP) and all files (\*.\*). The selection of MindMap application files is the default.



MindMap will default to \*.MM. If you saved your MindMap applications without the MM extension, you must explicitly type in the file extension or select "All files \*.\*" as the file type.

You can change the **Drive**: from where you want to access files or you can change the directory with a double-click on the symbol in the **directories**: list.

Select the desired file with a double-click on the file name, or by selecting the file in the list and clicking the OK-button.

## File | Save

If you have not previously saved the application you are working on, MindMap will prompt you for a file name. Any valid DOS file name (8 and 3 convention) is acceptable. It is suggested that you use the standard MindMap extension (\*.MM) for applications you wish to edit in the future. If you are running MS-Windows 95, you may also use the long file name feature.

This command saves the open file under the same name. The first time you save a file, a backup copy will be created with the extension \*.B00. MindMap initially creates up to three backups of every file. They are named \*.B00, \*.B01 and \*.B02 and represent the last three versions created by use of the **Save** command. Every time you save an application, a new backup is created and the oldest backup is overwritten. The oldest version is then stored in the \*.B02 file. You can load these files into MindMap and you are then able to see the progress of your application development. If the system abnormally aborts and you cannot access the MM version of your application, you can revert to a backup copy.

You can choose if you want no backup copies at all or if you would like to have automatic backup copies created by MindMap. This is set in the [File | Preferences](#) menu.

## File | Save as

If you want to rename an open file, you can use the command **Save as**. This option is also important for creating files other than MindMap application files. Since any file (*with the exception of EXEs*) created by MindMap is technically identical in its structure, only the extension will be a clue to its content and the manner in which you want it to be processed. If you save an application as an \*.MMP file, it will be used as a printer template. Saving the same file as \*.MM will permit you to subsequently edit and run it.

We strongly recommend using the suggested file name extensions.

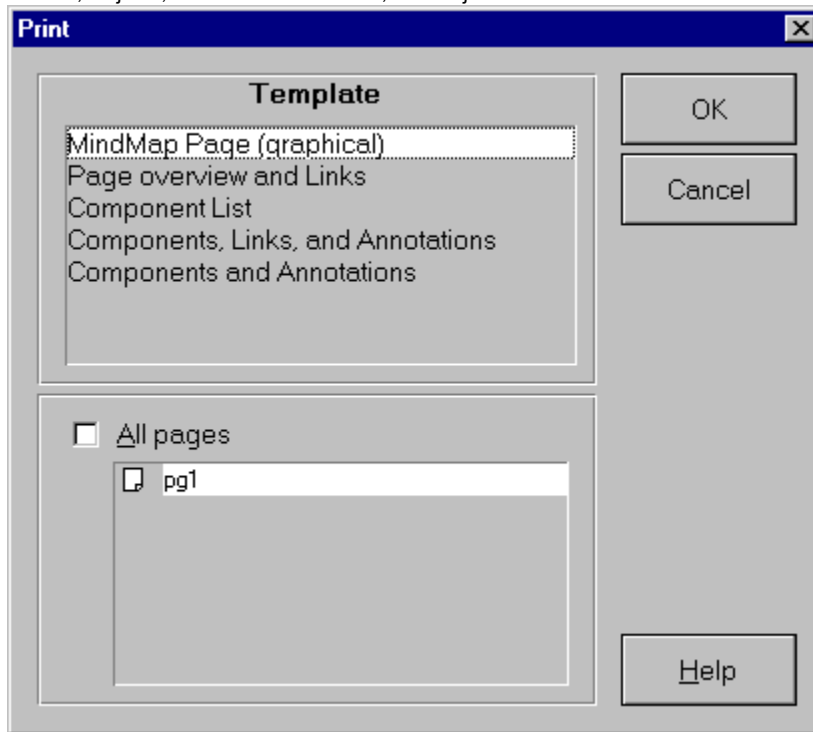
## File | Close

An open application can be closed with this command. It has the same functionality as the MS-Windows close button.

MindMap does not directly close applications that have been changed. There is always a warning and you are asked: "*Do you want to save the changes of (Application)?*". This warning gives you the option to save or not; e.g., correct changes or to reject undesirable changes made since you last saved the file.

## File | Print

The MindMap Print function allows you to look at five different layouts. It helps you analyze your applications and to create application documentation. The standard layouts are as follows: MindMap Page (graphical); Page Overview and Links; Object List; Objects, Links and Annotations; and Objects and Annotations.



## MindMap Page (graphical)

The option **MindMap Page (graphical)** opens a dialog box that specifies details of the standard printer. You can define the Print area ("*All*" or "*Pages from: to:*"), the "*Print quality*" or you can "*Print to file*". If you choose the option to only print specific pages, you must select the desired pages from the list. An option for printing only selected components is also available. The printout shows the components on the pages.



## **Page Overview and Links**

With this option, you can print the layout of the pages in a MindMap application. You will see the desired page with all components on it and with the component names. In addition, the printout will list the links you have defined for each component.

Before you print the Page Overview and Links, you can choose if you want to print only selected components, or all pages or only specific pages.

## Object List

**Object List** opens the same dialog box as printing a "*MindMap Page (graphical)*" and "*Page Overview and Links*". This printout shows a page oriented list with all of the components in an application. Again, you can choose if you want to print only selected components, all pages or specific pages.

## **Objects, Links and Annotations**

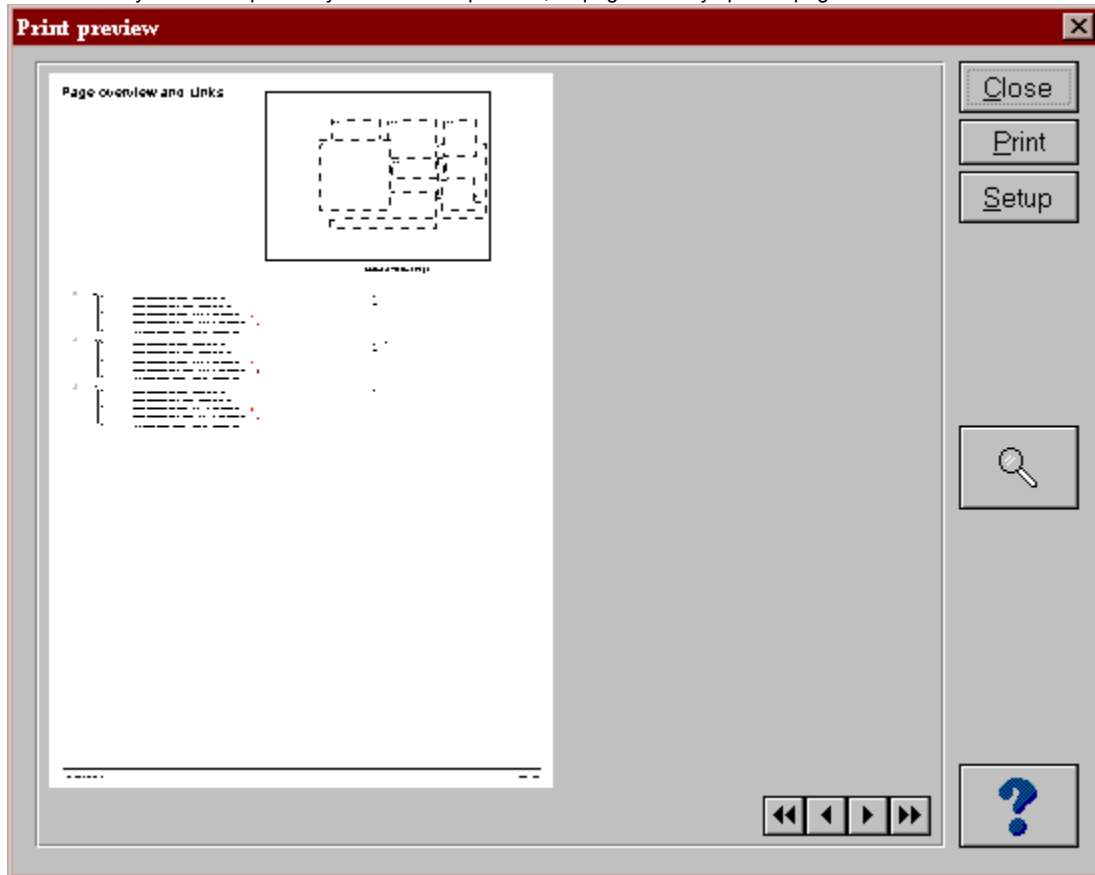
This option opens the same dialog box as the options before. The printout shows a page-oriented list of the components in an application. In addition, you will see the links and annotations of each component. Again, you can choose if you want to print only selected components, all pages or only specific pages.

## **Objects and Annotations**

This option opens the same dialog box as the options before. The printout shows a page-oriented list of the components in an application. In addition, you will see only the annotations of each component. Again, you can choose if you want to print only selected components, all pages or only specific pages.

## File | Print Preview

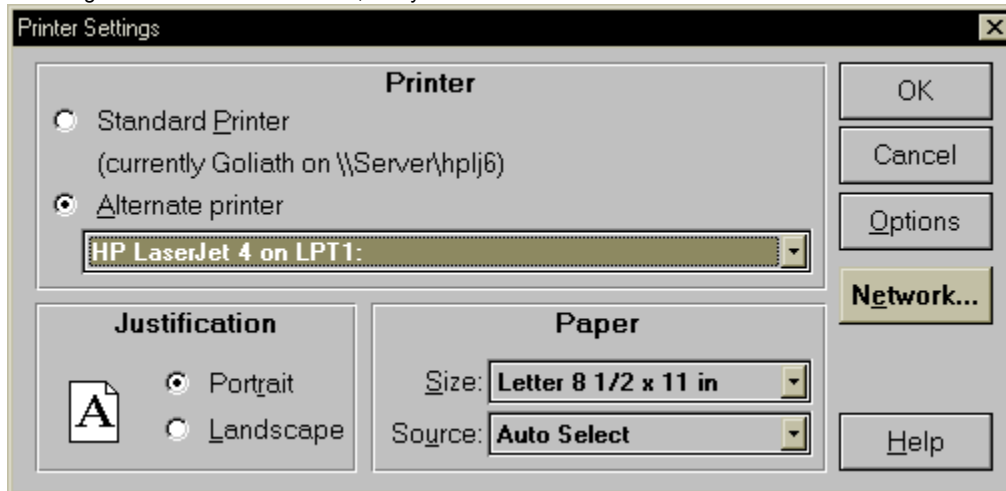
The MindMap "Print Preview" function allows you to look at the different layouts. It also helps you analyze your applications and create documentation, as does the **File | Print** function. The difference is that you will see a preview of the printed output on your screen. You can again choose one of the standard layouts: MindMap Page (graphical); Page Overview and Links; Object List; Objects, Links and Annotations; Objects and Annotations; or any you created yourself. In a dialog box, you can choose if you want to print only selected components, all pages or only specific pages.



The **Print Preview** dialog box consists of a preview of the screen you have chosen or of the first of several chosen pages. In addition, you can choose one of several options: you can close the preview dialog box; you can print the page; or you can change the standard layout you have previously chosen. A button with a magnifying glass lets you zoom in and out of the preview screen. Four buttons are at the bottom of the preview dialog box. If your selection has multiple pages, you can use the arrows to move to the first page, the previous page, the next page or the last page.

## File | Printer Setup

This command allows you to change the printer settings. You can choose the "Standard Printer" or an "Alternate printer". Select the appropriate orientation: "Portrait" or "Landscape" and the paper "Size" or "Source". Depending on your installed printer, additional options may be chosen. The standard message box for printer settings of your MS-Windows installation is being used. For more information, see your MS-Windows manual.

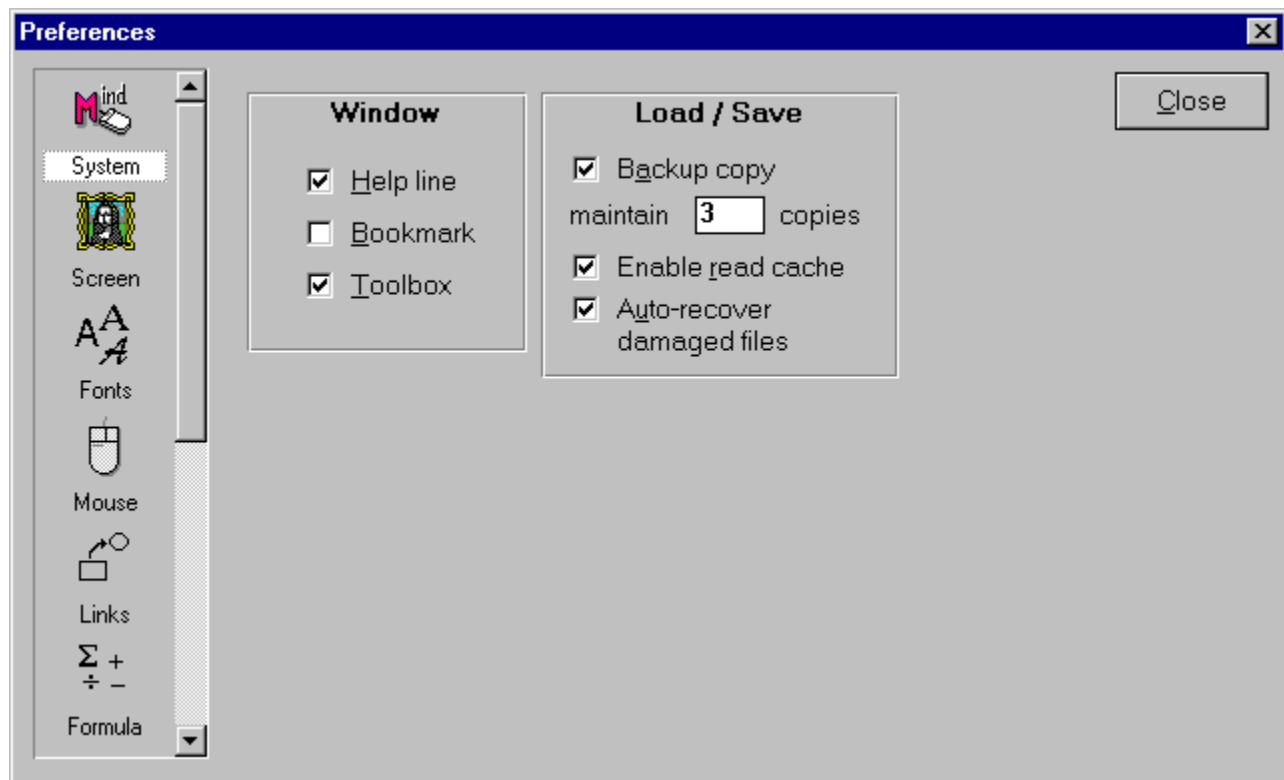


## File | Preferences

This menu selection offers access to numerous settings, enabling you to customize MindMap to your own specific preferences:

- [System](#)
- [Screen](#)
- [Fonts](#)
- [Mouse](#)
- [Links](#)
- [Formula](#)
- [Names](#)
- [Application](#)
- [Database](#)
- [Network](#)
- [Info](#)
- [VBX](#)

If you click on the menu command **File | Preferences**, a dialog box will appear which gives you these options to choose from. By clicking on one of the menu commands that appear in the list on the left side of the **File | Preferences** dialog box, you select which item to view.



Whenever you click on one of the items in the **File | Preferences** dialog box, new options will be offered for each selection. When you click on another item, without having closed the **File | Preferences** dialog box, MindMap will ask you if you want to accept the changes you have made or if you want to discard the changes. This is only true if you have made previous changes to other preference settings. The dialog boxes are described as follows.



## File | Preferences | System

In the "System" dialog box you can define whether a **Hint line**, a **Bookmark** and/or the [Toolbox](#) will be seen.

When you set the check mark of **Hint line** to "on", the **status bar** at the bottom of the MindMap screen will appear. The status bar displays component names, page scroll buttons, access to the parser window, the object list, and the field in which component names can be defined. If you switch the status bar off, you can still navigate to Previous page and Next page by using the *F7* and *F8* function keys. Alternatively, you can use the menu commands [Page | Next Page](#) or [Page | Previous Page](#).

If you have set the option, the name of the page where you have set a **Bookmark** will be shown in the right most section of the status bar.

The [toolbox](#), which contains the most commonly used components of MindMap, appears by default on the right side of the MindMap screen. With the appropriate check mark in the **System** dialog box or with the shortcut function key *F10* (or [View Toolbox](#) in the menu **View**), you can make the toolbox visible or you can hide it.

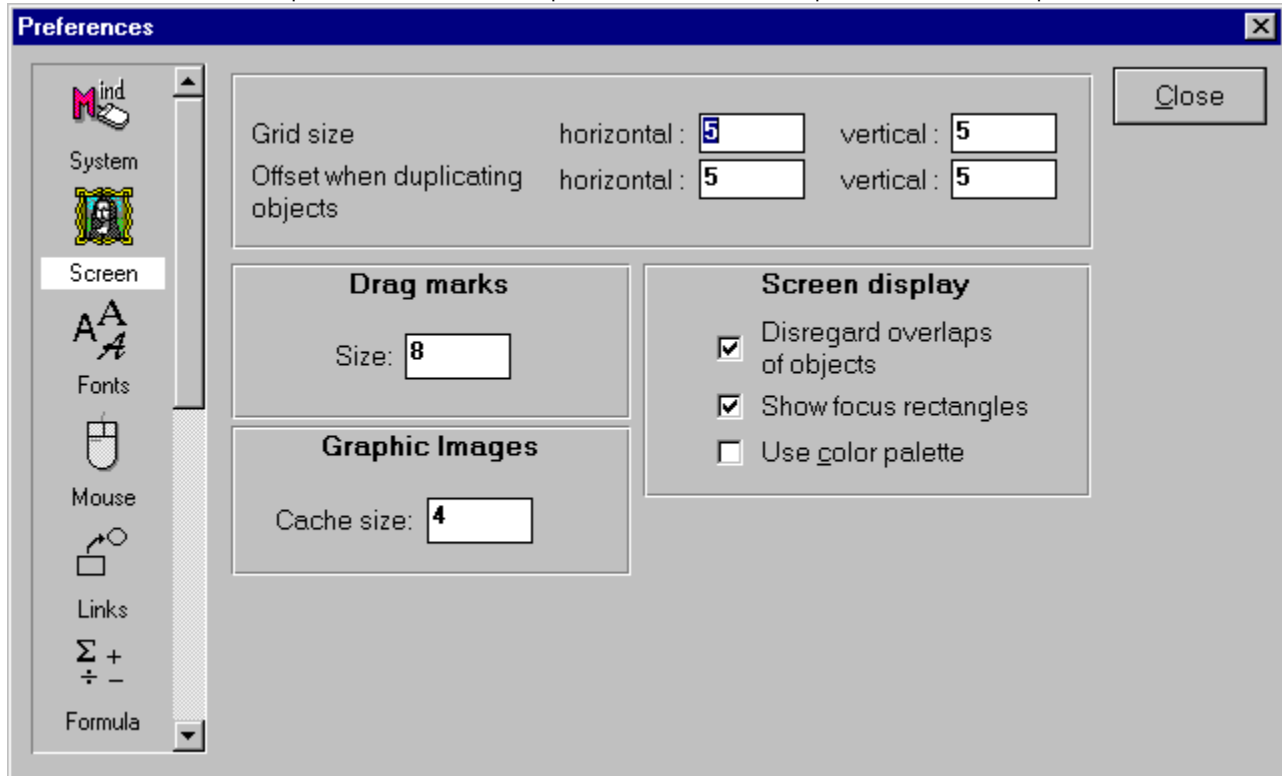
You also have various options regarding the loading and saving of applications. In the **Load** and **Save** menu, you can define whether MindMap is to generate a backup copy of your application whenever you save it, and if MindMap is to:

- maintain n copies
- enable read cache
- auto-recover damaged files
- swap temporarily

## File | Preferences | Screen

The **Grid size** determines the distance between screen positions where you can place a component when moving or creating components; i.e., when you specify "5" as grid size and you move a component it "jumps" by five pixels. The grid is invisible, yet allows you to more easily align components on the page. Note that, in most cases, it is not advisable to set the grid to zero.

The "Screen" dialog box allows you to define an **Offset when duplicating objects**. The default is five pixels horizontally and vertically. This allows you to create one component of the desired size, select this component and duplicate it as often as you want. Note that some components, like buttons and input fields, have their own implementations of the duplication offset.



**Drag marks** specify the size of the squares surrounding a selected component. Generally, you will not need to change this setting.

The setting **Graphic Images** allows you to set the number of images that are to be cached. This setting affects the speed at which graphics will be displayed on your screen.

**Screen display:** When you set a check mark for the option "*Disregard overlaps of components*", MindMap checks before painting any component on the screen. This is to determine whether all of it will appear or if some parts will be overlaid by other components. If parts of a component are hidden, they will not be painted. This leads to a smoother painting of the MindMap screens. Note that, depending on your video adapter, the performance of your computer can be a little bit slower with this option set to "on".

## **File | Preferences | Fonts**

You also have the option of selecting different fonts or font settings for various situations.

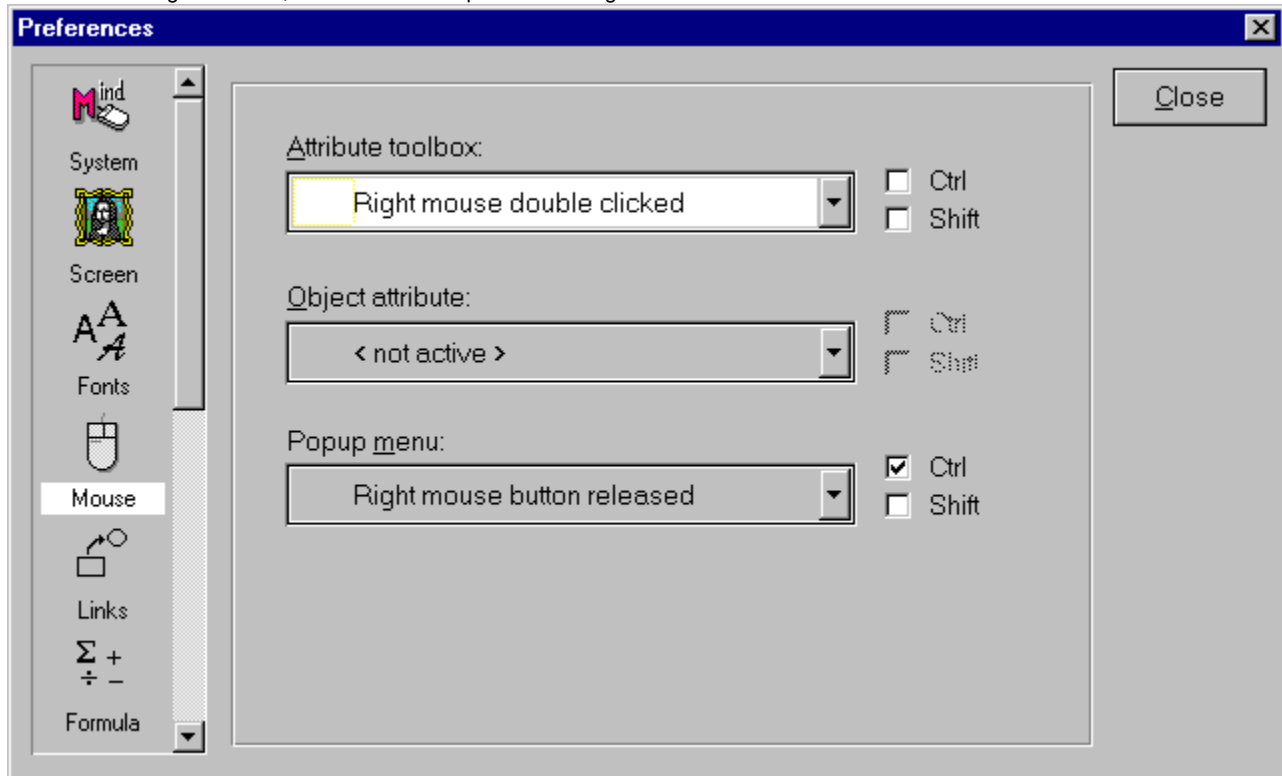
The Default Font setting will be used for the

The font selected for the object list will display whenever the objects are presented in list form, such as the tear-off list on the status bar, or in the link dialog boxes.

The font setting for the parser will be reflected in any input field where parsed statements can be entered.

## File | Preferences | Mouse

The "Mouse" symbol allows you to determine different kinds of mouse operations for opening the **Attribute Toolbox** of a component, the **Object attributes** or a **Pop-up menu**. You may configure the mouse behavior to your preference. Many MS-Windows programs have now standardized on accessing the properties of an object/component by means of right-mouse button clicking. Therefore, this is the MindMap default setting.



By default, MindMap will display the attribute toolbox of a selected component by clicking once with the right mouse button (this action is called "right mouse button released") on the selected component.

### Note:

Components that can have a focus - such as input fields, data tables, etc. - will not interpret a mouse click **inside their borders** as a command to display a selection. Clicking inside the border of such an object merely places the focus inside this component. If you wish to access the attributes, you must click **on the border** of the component. The mouse cursor will appear either as a flat hand or as a pointed index finger, when it is immediately above the border.

In the attribute toolbox of a component, you have the specific attribute. By default, MindMap does not offer a direct method to open this attribute by clicking on a selected component. Here you can change the settings to one of the three methods "right mouse button released", "left mouse button double clicked" or "right mouse button double clicked".

MindMap also offers the choice of opening a **Pop-up menu** which contains all items that are contained in the menu bar at the top of the screen. Here again, you can choose between the choices "right mouse button released", "left mouse button double clicked" or "right mouse button double clicked". This feature comes in handy, when you are in full screen mode and thus don't have access to the menu which is normally at the top of the screen.

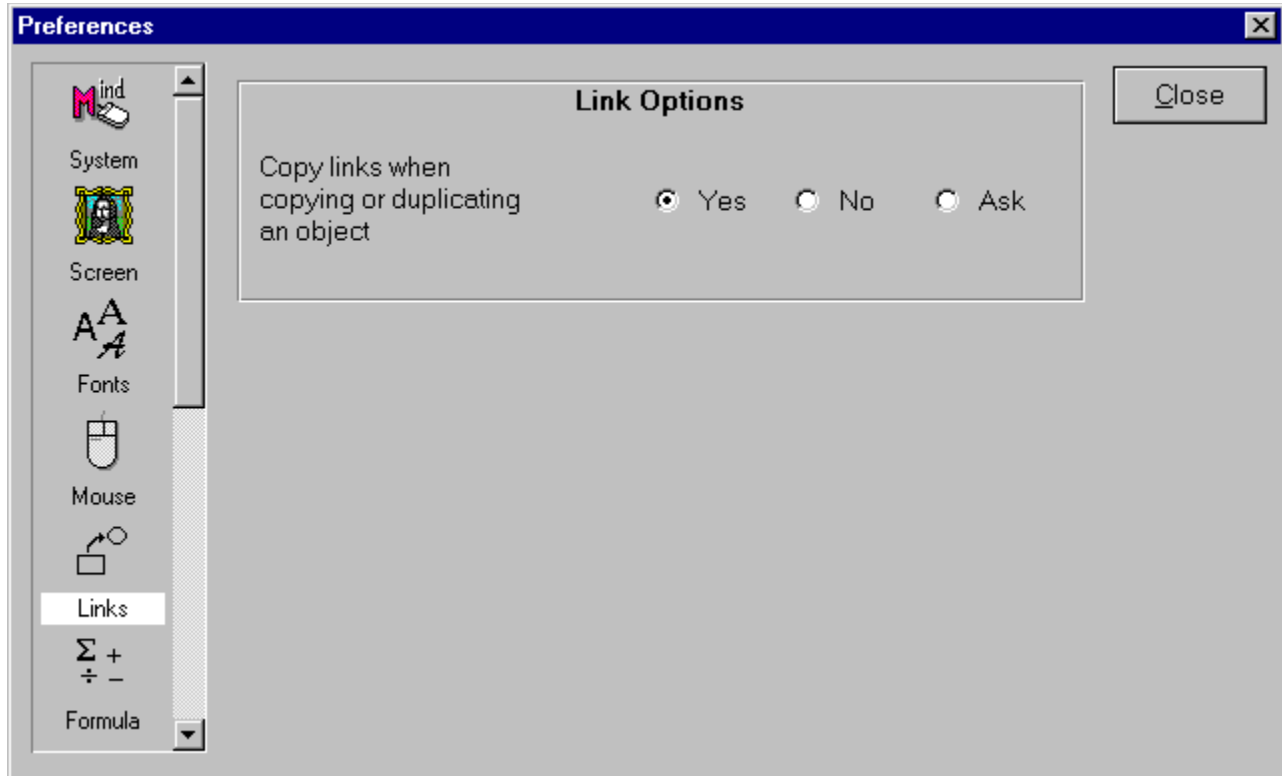
### Note:

Throughout this manual, the term "click" is used when an action with the mouse is being described. Depending on your settings of the **File | Preferences | Mouse** options, you have to choose the appropriate mouse action by yourself.

In all of the above mentioned cases, you have the option of prefacing a mouse click with either a **CTRL** or a **SHIFT** key. These modifier keys are offered so that you can distinguish between otherwise identical mouse clicks, although this does require a two-handed operation.

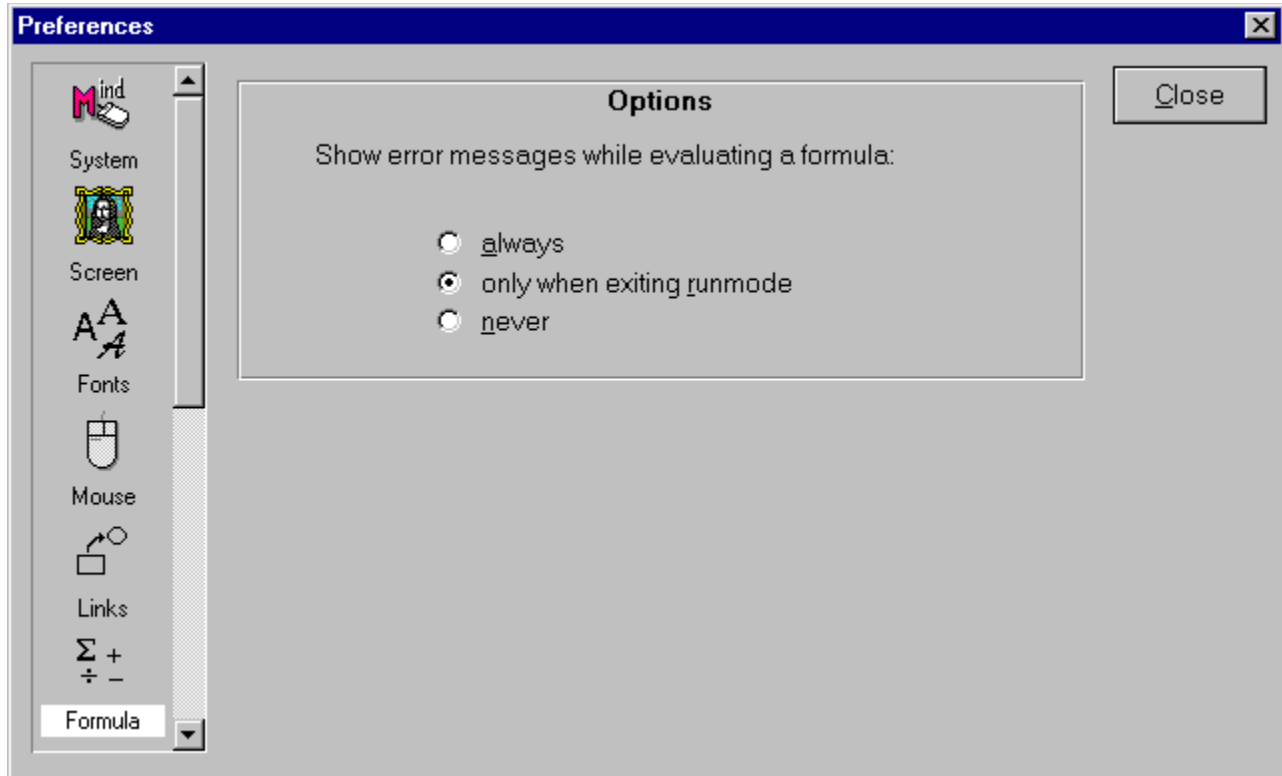
## File | Preferences | Links

The option "Links" allows you to choose whether links previously placed on components are to be copied along with the component when it is copied or duplicated. When you check the option "Ask", you will be asked every time you copy or duplicate a component, if you want to copy it's links as well.



## File | Preferences | Formula

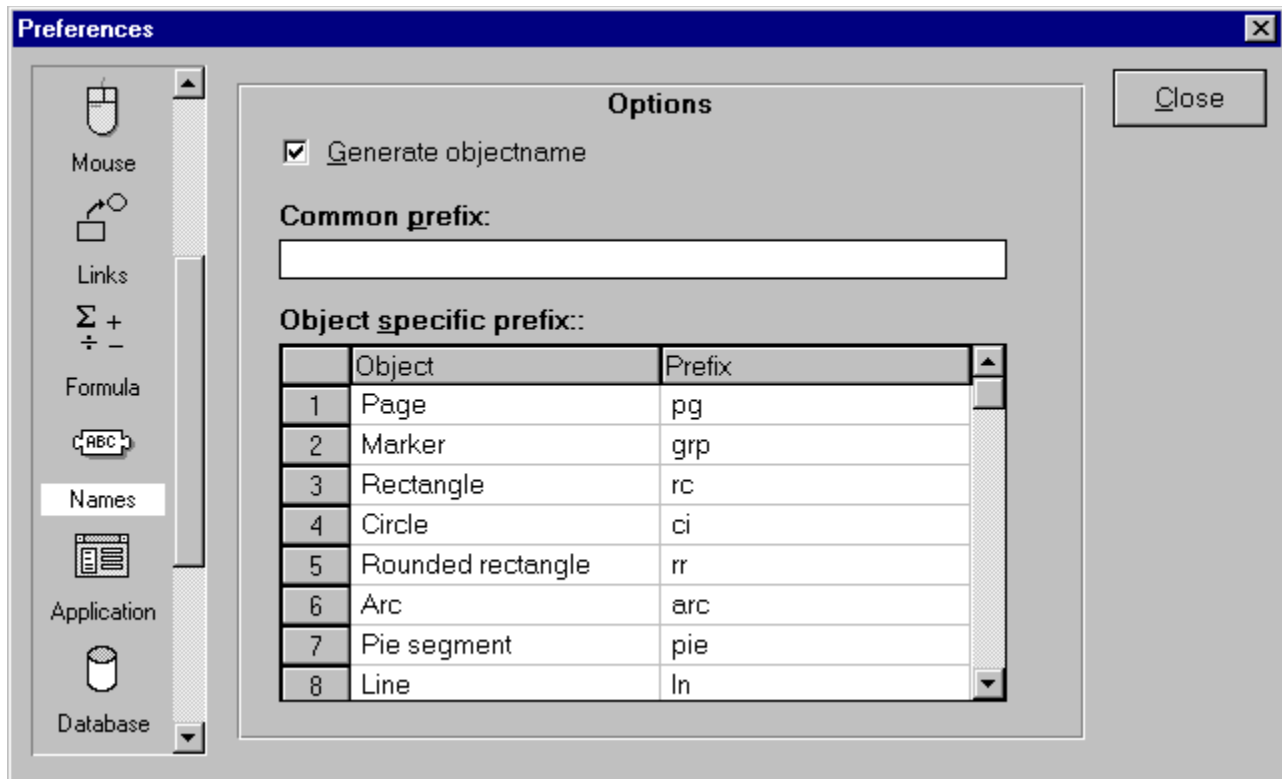
"Formula" has the option to **Show error messages while evaluating a formula** in three different ways: **always**, **never** or **only when exiting run mode**. You can view any errors that were detected by the parser as it was attempting to interpret the input.



When building an application, you might wish to have messages resulting from errors in conjunction with the parser to appear immediately when encountered (**Always**), or never (**Never**), or only when terminating the application. If you choose the first option, a dialog box displaying the error message will appear during execution. You can either disregard it and deal with it later, or you can interrupt the application and attempt to fix it. If you choose the option **Only when exiting runmode**, a dialog box will be displayed at the moment you terminate the application. The dialog box will then display a list of all error messages encountered during the execution of the application.

## File | Preferences | Names

While developing your application, it is useful to assign self-explanatory names to the different components.



The preference "Names" is designed to give different components a special prefix so that one can differentiate them; e.g., you can define each command button to have the prefix "btn". After creating your buttons, they are named "btn1", "btn2" and so on. To enable this feature, the check box "generate object name" must be checked. You can assign a common prefix for all components by typing the desired letters and special characters into the corresponding text dialog box. This appends your prefix between the component type name and it's incremental number. E.g., if you type in the letters "\_name\_", your buttons will get the names "btn\_name\_1", "btn\_name\_2" and so on. This feature is particularly useful in team application development or maintenance.

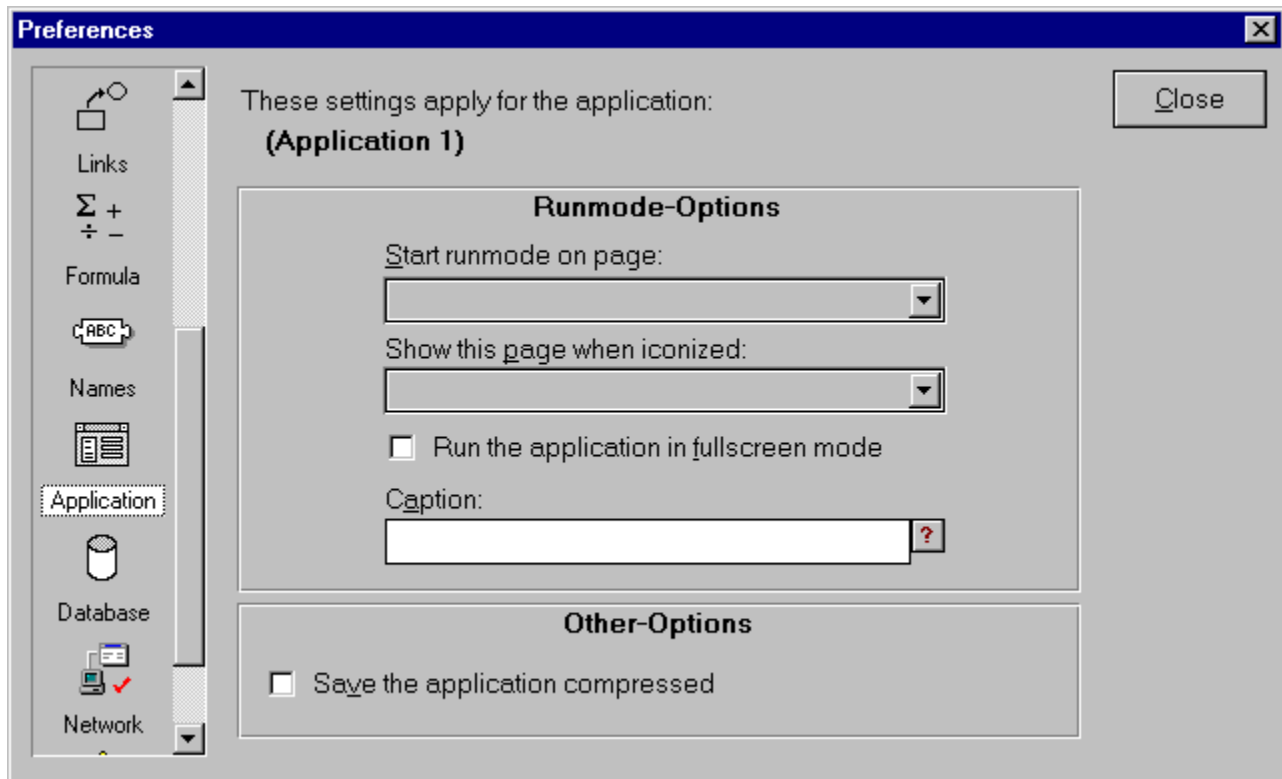
MindMap is shipped with a default set of names for all components. It is strongly recommended that you keep this default setting. In addition, it is also suggested that you always give a component a name which hints at its purpose, relationship to other components, or whatever. This makes navigating the application space much easier and it will assist you, or others, in maintaining the application.

### **Note:**

*MindMap requires that a naming convention be considered when assigning names to components. A component name may have a maximum of 255 characters. It should not include special characters such as blanks, question marks, etc. You should not use numerals as names. The reason for these limitations is the parser operation. If a component were to be named '1' and another one be named '2', then the parser would not know whether the statement 1+2 refers to the values of the components named '1' and '2' or to the numbers themselves. You are permitted to use any conceivable name, but MindMap warns you when you attempt to disregard the conventions.*

## File | Preferences | Application

When you change from edit mode to run mode, MindMap starts running the application, by default, at the first page of your application.



The preference setting "*Application*" allows you to specify a page that functions as the starting page in run mode: **Start run mode on page**.

This facility is used if you intend to put pages at the beginning of your application which are to remain inaccessible or invisible to the user. Typically, such pages are backgrounds, locations for general purpose components, configuration pages, etc.

As an easy way to test a special function on a page that is not the first page of your application, just tell the program to start with that page. This avoids having to go through your entire application to only test one function or page. You can then easily re-set the start page to the original start page or another page you want to test.

Choosing the option **Run the application in full screen mode** hides the title and menu bar. This setting is generally used when building multimedia applications, which are not to include typical MS-Windows menus, caption bars, and/or status bars.

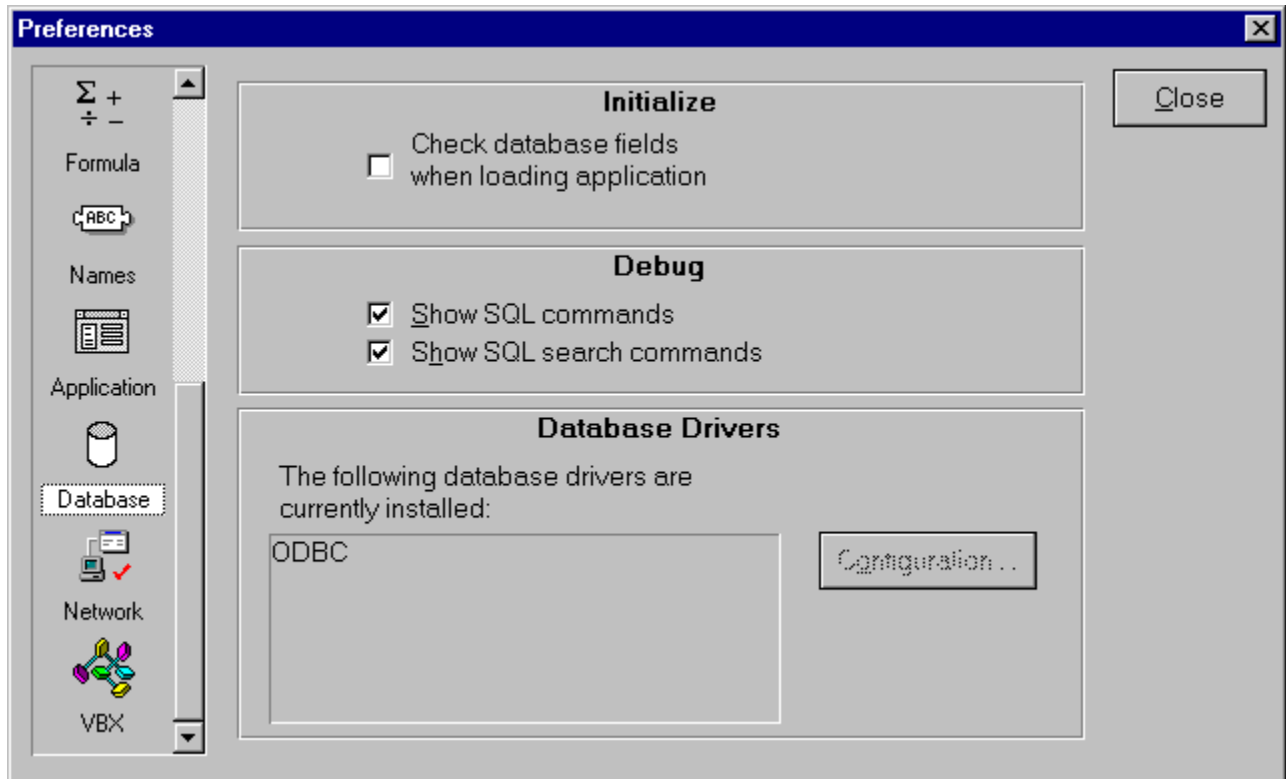
If you want to **Save the application compressed**, choose the special check box in that dialog box. MindMap then saves space on your hard disk by compressing the application. The compression algorithm MindMap employs is a compromise between minimal file size and minimal loading time. Higher compression leads to longer loading time, since the file must be decompressed. Feel free to use an external compression utility, since there definitely will be more space you can save, but you must be aware of possible performance degradation.

If you select one page of your application under the preference "*Show this page when iconized*", a portion of this page is shown in MS-Windows, when the user minimizes your application in run mode. You can use this facility to create your own desktop icons for MindMap applications. Be sure to place the bitmap at the top left hand corner of the page you have designated. Only the extreme top left 32 x 32 pixels will be shown. Make sure that the bitmap corresponds to the size requirements for desktop icons (32 x 32 pixels). You might also want to experiment with altering icons, which could be used to display various states of your application. This feature only applies to MindMap installations running on MS-Windows 3.11 or MS-Windows for Workgroups.



## File | Preferences | Database

The "Database" preference allows you to **Check database fields when loading the application** during the initialization. The purpose of this is to



Another option allows you to display SQL commands sent to the database driver. You can elect to **Show SQL commands** and/or to **Show SQL search commands**. This is very helpful in determining the validity of SQL commands that your application is issuing. When you dynamically construct SQL commands (via the parser), the database driver may not actually be receiving what you think it should receive. You can also see the actual SQL commands MindMap generates. Selecting one of these options will make the process more visible to you for analysis.

To see what is happening, you need to set the option [View System Log](#) in the **View** menu to on. You can also use the shortcut **CTRL+F10** to switch the MindMap System Log on or off. In the MindMap System Log dialog box, you can view every SQL message.

Yet another way to look at the SQL command messages is to look into the file MMERROR.LOG. MindMap writes all SQL messages into this file, provided that you have set the "Logging Level" in the MINDMAP.INI file to "4" or greater. You can determine whether this has been set by opening the MINDMAP.INI file and scrolling to the entry, System. There you will find the following statement:

[System]

Logginglevel = 4

In addition, the following entry must also be set:

[ODBC]

trace=1

The valid entries are:

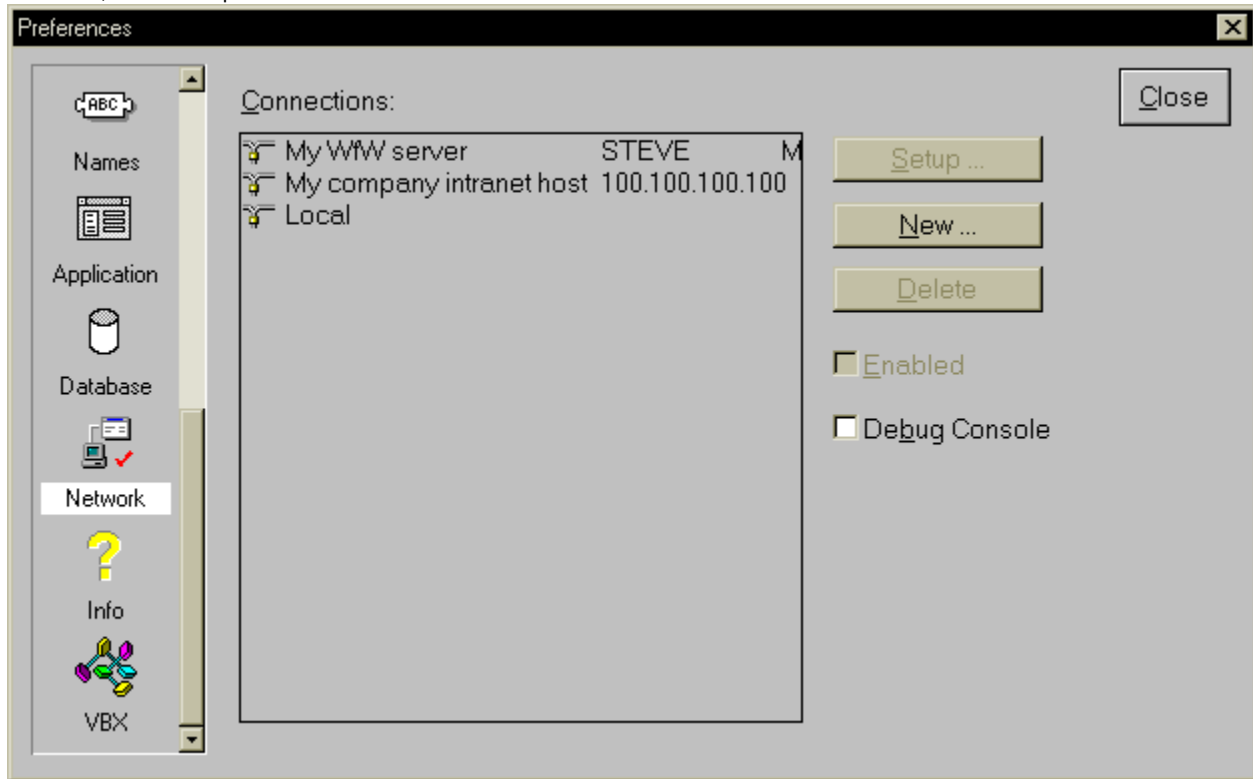
LOG_FATALERROR	1	Fatal errors such as "Unrecoverable Application"
----------------	---	--

#### Errors“

LOG_ERROR	2	MindMap errors ( <i>e.g. Parser statements</i> )
LOG_WARNING	3	MindMap warnings ( <i>e.g. Parser statements</i> )
LOG_MESSAGE	4	Information ( <i>e.g. from the database manager</i> )
LOG_TRACE	5	Statistics
LOG_WINDOWS	6	Internal MS-Windows warnings

## File | Preferences | Network

This option allows you to define the various network connections to the servers. The servers here are meant in the physical sense, i.e. the computers to which a connection is to be established.



A detailed discussion of these settings can be found in the chapter dealing with client / server technology.

## File | Preferences | Info

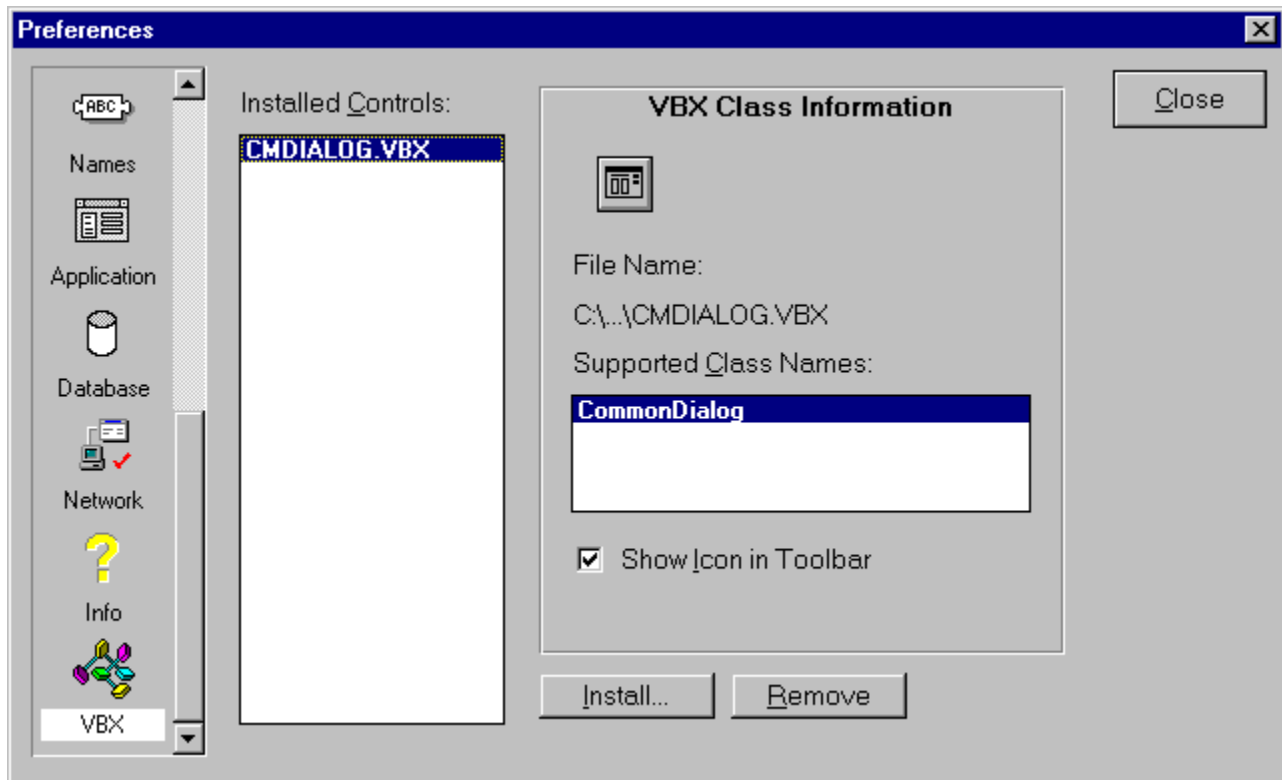
In the preference menu, general information is available. **Info** gives you information about the libraries for the different MindMap components.

You can view an [Object list](#). Clicking this option outputs a list of all components of the current application to an external text file. You are prompted to key in the name of this text file and will then be asked if you want to review it immediately. If you indicate **Yes**, MindMap automatically loads the file in a standard MS-Windows word processor, such as "Notepad".

The **Global Memory**, the **Error Log** and the **System Heap** can be viewed and output. This can be used for support purposes, such as when you use our support system. The contents of the message boxes that will be generated, can be forwarded to the MindMap support staff.

## File | Preferences | VBX

The **VBX** preference allows you to **Install** or **Remove** Visual Basic controls. This is a powerful facility that gives you access to a huge library of Visual Basic controls or eXtensions (VBX). You can quickly and easily integrate new features into MindMap with this option.



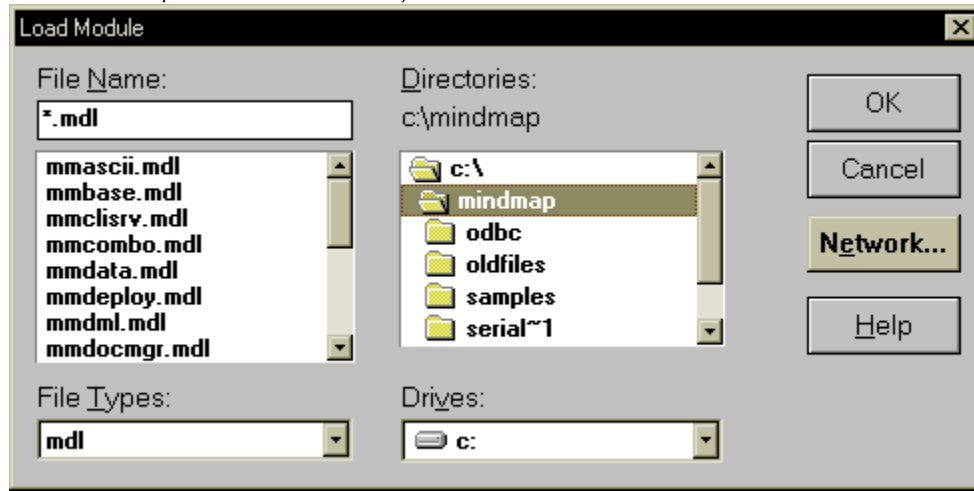
To install a VBX component, click on **Install** and search for the desired file with the extension \*.VBX. After choosing the VBX component, its name is shown and the **Supported Class Names** are listed. If there is an icon representation of the VBX available, and you wish to use it, check this box. If you do not want to have the icon appear on the toolbar, then do not check this option. It will still appear in the menu item **Objects**. MindMap must now be restarted, so that all modifications will take effect. After the restart, providing you chose to have an icon for the VBX, an icon in the toolbox will symbolize the VBX component and you can work with it, as with any other MindMap component. The VBX component is also listed in the **Objects** menu of the menu bar.

You can remove a VBX control as easily as you can install it. In the "Installed controls" list box, highlight the name of the VBX component that you want to remove in the "Installed controls" list box and click on the **Remove** button. MindMap then asks you if you are sure that you want to remove this control. If you click on the "Yes" button, the control will be removed. Note that this modification also becomes effective only after you have restarted MindMap. Note that removing the VBX from MindMap does not delete it from your system.

**VBX implementation notes:** *There are certain limitations to which VBXs can be installed in MindMap. Any VBX requiring facilities built into Visual Basic can obviously not be used, since Visual Basic is not installed or running inside of MindMap. Currently, Microsoft only allows third party environments (such as MindMap) to support VBXs which comply with the VBX 1.0 standard. Thus, only those VBXs which support at least VBX 1.0 specifications can be installed in MindMap. You must determine the compatibility by either reading the documentation supplied with the VBX or by contacting the vendor. Also note that MindMap has no method of knowing or modifying any behavior of a VBX, other than what the original developer of the specific VBX has allowed. E.g., if they provide the capability for you to change a color in the VBX, then MindMap will allow you to do so. Also, when you deploy your application that incorporates the functionality of the VBX, the VBX must be deployed with the application, which means you must have the necessary license to do so. This is beyond the purview of MindMap and must be dealt with between the individual developer and the owner of the VBX.*

## File | Load Module

MindMap is a module oriented program. When you start MindMap, numerous MindMap modules are loaded. (*You can observe the process in the status bar.*)



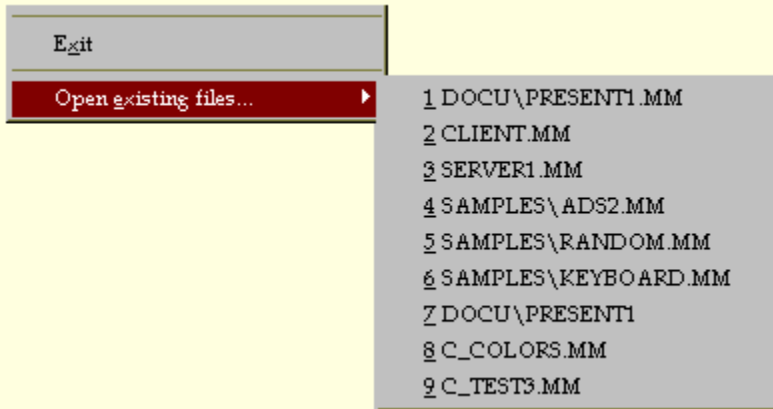
Whenever you want to load additional modules, choose the option **Load Module** in the **File** menu. You then select a file with a \*.MDL extension that loads the appropriate module. This facility is only necessary when you wish to load a MindMap module (MDL file containing components), after MindMap has been loaded initially. In general, MindMap installs all modules (MDLs) it finds in its default directory and those specifically pointed to in the MINDMAP.INI file.

## File | Exit

If you want to leave MindMap, choose **Exit** in the file menu. MindMap will prompt you to save your application if you either made, or attempted to make, any changes while the file was open.

## File | Open existing files

This command is only available if you have previously saved any MindMap files. The most recently saved file stays on top of the list. Up to nine files are listed and can be chosen very quickly. A "More" option shows additional older files.





## Edit Menu

Since editing is one of the most common tasks in application development, we will offer some general information about the selecting, sizing and dragging of components. After this, we will describe the items that you can find in the **Edit Menu**.

## Selecting Components

A number of actions dealing with components can only be performed if the component has been selected. Clicking with the pointer on a component selects it. The component will receive a selector indication, i.e., a hatched border with rectangles (see also *File | Preferences | Screen | Drag marks*) in the corners and along the borders of the component.

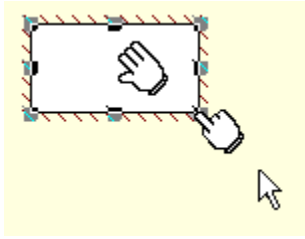


In some cases, you can click anywhere on the component in order to select it. If the component is a type which accepts a focus in run mode (input fields, radio buttons, etc.), you must select the component by clicking on its border. The color which is used to draw that border can be set by the MarkerColor keyword in the [System] section of MindMap.INI. It uses RGB format. It is red (255,0,0) by default.

You can also select components in other ways:

1. Hold the left mouse button down and move the mouse cursor around one or more components by starting somewhere outside the lower right of the component or group of components and moving somewhere to the upper left of the component or group of components. You don't need to move the mouse cursor totally around the component or group of components - just "touching" them is enough to select them. Once you have finished selecting the desired components, release the mouse button. You will then see a selection marker around the entire group. This is the fastest method to select a group of components.
2. The second method is to move the mouse cursor from outside of the upper left of the component or group of components to the lower right of the component or group of components. Once you have finished selecting the desired components, release the mouse button. You will then see a selection marker around the entire group. In this case, only components that have been enclosed completely by the mouse cursor are selected.
3. Another way to select several non-contiguous components is to hold down the *SHIFT*-key and select the desired components by clicking on them with the mouse cursor.
4. In addition, you can select a component by using the Object List. Expand the list, navigate the list until the name of the component to be selected is visible. Now press the *SHIFT* key, keep it pressed and click on the component with the mouse. See the explanation in the section *Edit | Go to Object* and especially the note in *Component List*. This is especially useful in locating small or otherwise hidden components in your application.
5. If you select components using the methods described in 1) or 2) above, then a temporary group will be created. If you now attempt to access the attributes, you will notice that the set of available attributes might have changed, depending on the individual components contained in the group.

## Sizing Components



The selector rectangles around a component have different functions. If you move the pointer into one of the drag marks, the cursor will change from an arrow to a hand with a pointing index finger. This means that you can now enlarge or shrink the component by pressing the left mouse-button and dragging the component to the desired size.

If you drag the component by a rectangle in the middle of the border, the component changes its size horizontally or vertically. If you drag the component by a rectangle in the corner of the border, you will change its size: either horizontally or vertically or both. In this case, when you also press and hold the *SHIFT* key, the component will keep its length / height ratio.

By using the corner rectangle, a component can get a rough shape and with the middle rectangles, one can precisely define the height and width of a component.

**Rectangle - rc1 - (430,250) - [170 x 150]**

The status bar can be very helpful in determining the exact size and location of a component. The status bar states the class (*component type*) and the name of a selected component. It also displays the position of the component in rounded brackets and the size in square brackets (*horizontal and vertical position in pixels*). The position is defined by the upper left pixel of the component. (See also the parser functions [xPos](#) and [yPos](#).)

## Dragging Components

Whenever the pointer moves into a selected component, the cursor changes to a flat hand. (*If the component you are dealing with can receive a focus, then the cursor will only appear as a flat hand, when it is immediately above the border of the component.*) The flat hand symbolizes that you can drag the component by pressing the left mouse button. Releasing the mouse button will leave the component in its current position.

## Edit | Undo / Undo not available

MindMap is capable of Undoing most actions relating to the drawing of components. Any operations to the file system, to the printer, or in conjunction with any other external devices, cannot be undone. Since the history of all actions must be stored in local memory, there are also some limitations as to how far back you can undo operations. You also cannot undo operations relating to attributes and links.

After you start MindMap, the command **Undo not available** will appear in the Edit menu. From then on, most actions will be recorded and can be undone.

## Edit | Cut

This is a common MS-Windows command. It is used by MindMap in the same manner as in most other programs. You can cut and paste components between different MS-Windows programs, within a MindMap application or between different MindMap applications. If you are not familiar with this command, please remember the shortcut key *CTRL+X* for cutting, as well as *CTRL+V* as the shortcut key for inserting. The cut command copies selected components into the MS-Windows clipboard, where it can subsequently be used for different purposes. It also deletes the original component from MindMap. (See also [Edit | Copy](#).)

MindMap places the components onto the clipboard in various formats:

Component	Clipboard Representation
-----------	--------------------------

Graphical Primitives	
----------------------	--

Button, Check boxes, Radio buttons, scroll bars	
---	--

Graphics	
----------	--

## Edit | Delete

**Edit | Delete** deletes selected components. If you have been too quick with this command, don't worry - the Undo command will help. Another possibility to undo your actions is to exit a MindMap application at this point without saving. For example, imagine you are developing an application and you have saved it as a file with the name DEMO.MM. You then proceed to make alterations to your application only to realize, that they weren't such a great idea after all. If you exit the application without saving it, you can then revert to the previously stored version to regain your initial position.

If you have links on the component (*or components*) that you are trying to delete, MindMap prompts you to confirm that the components have links to other components that will be permanently deleted. The **Edit | Undo** command will not be available to restore the links, once deleted.

## Edit | Copy

The **Edit | Copy** command copies a selected component into the clipboard. Whenever you use the [Edit | Paste](#) function, the component that has been copied will be placed at the same screen position it was copied from, but only on other pages than the original. The component is stored in the clipboard as long as no other component replaces it. This command can be used to place identical components at the same position on different pages (*e.g. titles or navigation buttons*), or in different MindMap applications. It can not be used to place a component on the same page as the original.

In addition, you can not only copy components, but also the values of components. If you highlight the displayed value of an input field that shows a text, you can copy that text to another input field or another component, by cutting & pasting it or by copying & pasting it. Please note also that the MS-Windows clipboard acts in different ways if you copy components and links. Most components are copied in MindMap format (*for pasting the copied component to another position in MindMap*) and in MS-Windows metafile format, enabling you to paste the component into other applications. Links are copied in MindMap format (*for pasting the copied link to another position in MindMap*) and as a bitmap file for pasting the link to other applications.



## Edit | Paste

The **Edit | Paste** command copies the content of the clipboard into the current page of the active application. For more details, see the [Edit | Copy](#) function or the [Edit | Cut](#) function. Also note that the shortcut key *CTRL-V* will not paste on a page other than the original page from which the copy was made.

## Edit | Duplicate

The **Edit | Duplicate** command copies selected components and places them on the screen, with a predefined offset.

Command buttons, check boxes and radio buttons are duplicated vertically; i.e., they are located in a column, so that they are placed correctly. The spacing can be changed in the [File | Preference | Screen](#) menu. The other components are duplicated with a spacing that is fixed. You can change this spacing with the function **Offset when duplicating objects** in the [File | Preference | Screen](#). The difference between [Cut/Paste](#) and **Duplicate** is that the former copies a component at the same position (*via the clipboard*) and the latter copies a component in a new location (*slightly off-set*). You can also duplicate a selected component by using the *F2* key.

Note that some components react differently to duplication operations (*F2*):

- **Command buttons:** The new push-button is created below the original with the vertical grid displacement in between.
- **Check boxes and Radio buttons (*normal style*):** The new component is created immediately below the original with no horizontal displacement.
- **Check boxes and radio buttons (*push-button style*):** The new component is created to the right of the original with no displacement.

A duplicated input field appears below its original with no displacement, but note that a small displacement appears because child windows are created in an inflated rectangle (*by a size of 4x4 pixels*).

## Edit | Copy Links

The **Edit | Copy Links** command is useful when copying recurring links, e.g., copying cursor changes as a result of mouse movements into or out of components. To copy links, select a component that contains the corresponding links and copy the links by clicking on that command. Now select the component that is to receive the links and open the [Link](#) function in the **Properties** menu (*shortcut key F6*). Click on [Edit | Paste](#) and the links will be pasted onto the selected component.

## Edit | Append links automatically

When developing a MindMap application, you will sometimes desire to use the same links in multiple locations. Before placing or copying all these links individually, use the convenient command **Append links automatically**. A dialog box will appear. In a list, you select the component type that will get the links, e.g., a command button. Then click on **New** and select the appropriate link, e.g., "when mouse moves into component change cursor". Before you assign the links, you must choose if this command applies to the components of **all pages** or to those of the **active page only**. The **Start** button executes the command immediately.

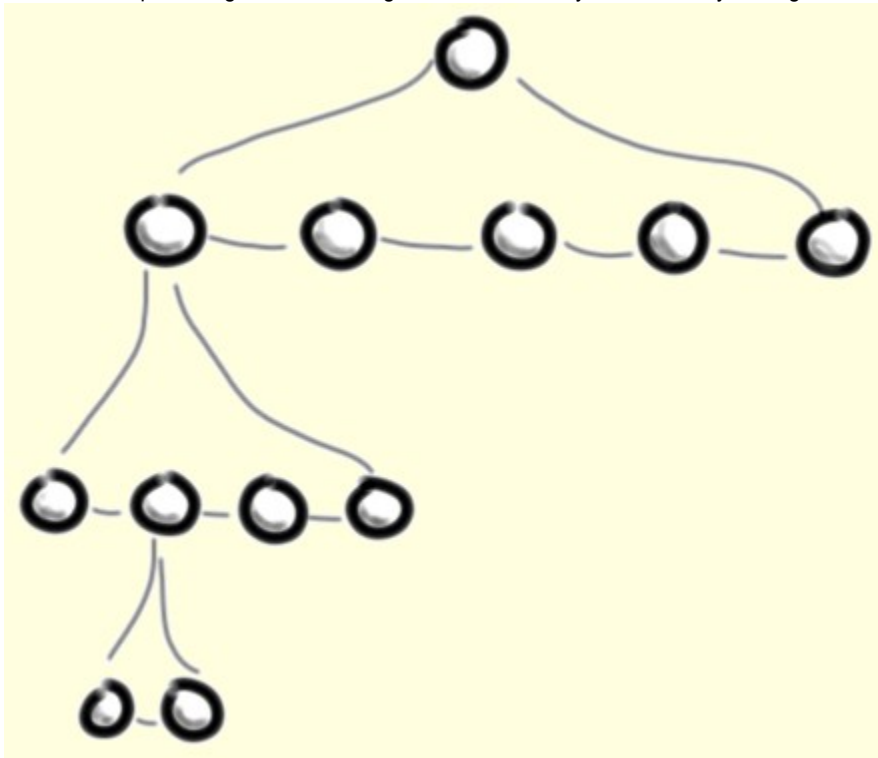
If you have set a check mark to "confirm all actions" and you click the **Start** button, MindMap jumps to the first component. Now you click on "Append" to append the link to this component or you can click on the "Next" button, such that MindMap jumps to the next component without appending the link. In this manner, you may be selective about which components receive the links.

## Edit | Foreground

If you have more than one component on a page, some of the components may overlap. MindMap places the components in the **Foreground**, depending on their initial order of creation. In other words, those components you create first on a page are placed in background, the next are layered on top, and so on. The last component placed is on the top layer - it is in the foreground.

You can change the sequence of the components, and the order in which MindMap will evaluate them in run mode, by assigning the components to foreground or to background. If you move a button onto a picture, the button will disappear if it was created before the picture. If you select the picture and place it in the background, the button will then appear. The order in which you have placed components and the subsequent rearrangement of components in fore- and background respectively, determines in which order MindMap paints the screen and in which order you can tab to components that receive focus. MindMap works from background to foreground and thus from first created components to later created components. In other words, the order in which MindMap looks through each component, if there is a link to be evaluated, is from background to foreground.

The concept of foreground and background is more easily understood by looking at the following graphical representation.



The top pearl in this chain represents the application itself. The next level, containing 5 pearls, represent the pages of the application. The third level - with four pearls - symbolizes 4 components on the first page. The second component from the left, is a group component, which consists of the two components displayed immediately beneath it. When the user initiates an event (*a mouse click for example*), the application is the first entity to receive it. The application then forwards it to the first page. The first page then determines if components have been placed on the page. If this is the case, then the event is forwarded further down the chain into the first component. If the message is not intended for this component, it is passed on to the next pearl in the chain. In this example, the next pearl is a placeholder, a group component. It takes the event and passes it down to the first member of the group. If both of these don't know how to deal with the event, it is forwarded back up to the component level and passed to the right.

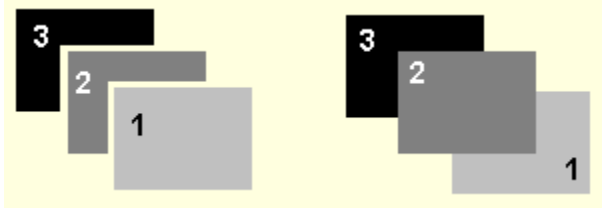
Event passing occurs from left to right. The leftmost components are those in background. The rightmost components are those in foreground. Moving a component from foreground to background is actually rearranging the sequence of the pearls on the chain. Creating a group is really inserting a new component (*which does not have a visual representation*) and dropping the affected components down one level.

With this in mind, you can control how the application evaluates the components of a page looking for links, page after page.

See also [Edit | Set Tab Order.](#)

## Edit | Background

The **Edit | Background** and [Edit | Foreground](#) commands change the order of the components. Selecting a component and then assigning it to background, makes it the first component to be painted and the first to be processed, assuming links are associated with it. Conversely, moving a component to foreground will make it be painted and processed last.



In the first diagram, component number 1 is in foreground. If selected and placed in background (**Edit | Background**), it will be 'behind' component Number 3, as shown in the second diagram. It thus will be painted first and also processed first.

## Edit | Go to Object

The shortcut function key *F5* or the **Edit | Go to Object** command allows you to jump to a specific component. You can jump to any component on any page of your application (or to any other *MindMap* application which is open at the time) and select it.

### **Note:**

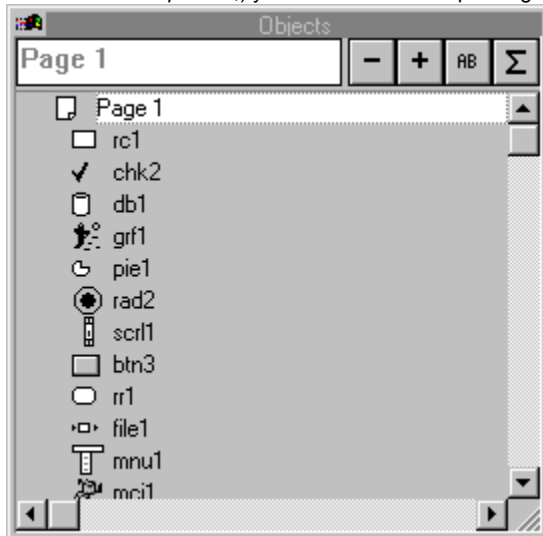
*This is another reason why you should use self-explanatory names when naming your components. MindMap names the components automatically by setting a counter after the name prefix, but if you name the important or key components individually, application development is easier because you will recognize them faster.*

Pressing the shortcut function key *F5* opens a list with all the components in your program. Now you can choose any component to which you want to go. The list shows all components of all applications which are open, but you can set a filter for specific component types, e.g., for a definite command button. Select the appropriate component in the list **Object type**.



## Component List

Another way to quickly jump to any of the components in the currently active application is to use the component list at the lower right corner of the MindMap screen. Just to the right of the text field which shows the name of the page (*or of any selected component,*) you will see an arrow pointing up. Click on this arrow and the list with all components will appear.



You now have several options:

- clicking on the '-' sign shows only page components
- clicking on the '+' sign shows components of all types, ordered by pages (*the default view*)
- clicking on the AB sign sorts the component list alphabetically. Now you can enter any letter, or combination of letters, in the white text field to the left of the - sign and the component list will show components whose names start with the letters you have keyed in.
- clicking on the little collection of rectangles will display the components in the order of creation by pages.

When you find the name of the component you are looking for in the component list, press down the *SHIFT* key and click on the name of the component. MindMap will jump to the page where the component is placed and will select the component. Instead of pressing the *SHIFT* key and clicking on the name of the component, you can also press the *ENTER* key while the *SHIFT* key is pressed.

### **Note:**

*If you drag the components list a little bit away from its original position, it will stay on top of the MindMap screen until you close MindMap or until you close the components list. This allows easier navigation within your application.*

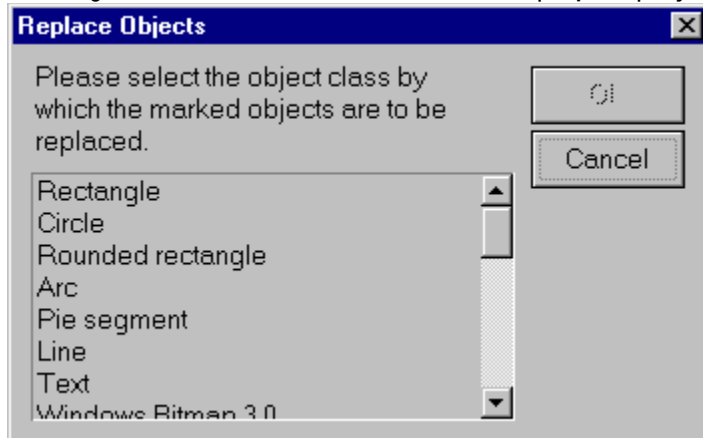
## **Edit | Replace**

Sometimes it becomes desirable to change either components or attributes of components on a global basis. In other words, instead of editing each individual instance of a component, you can make a replacement for all instances of a component. There are two options:

- you can either replace components or
- attributes of components.

## Edit | Replace | Object

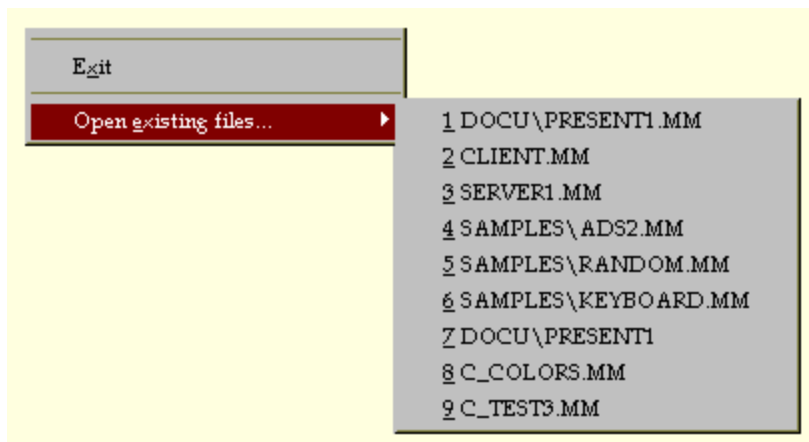
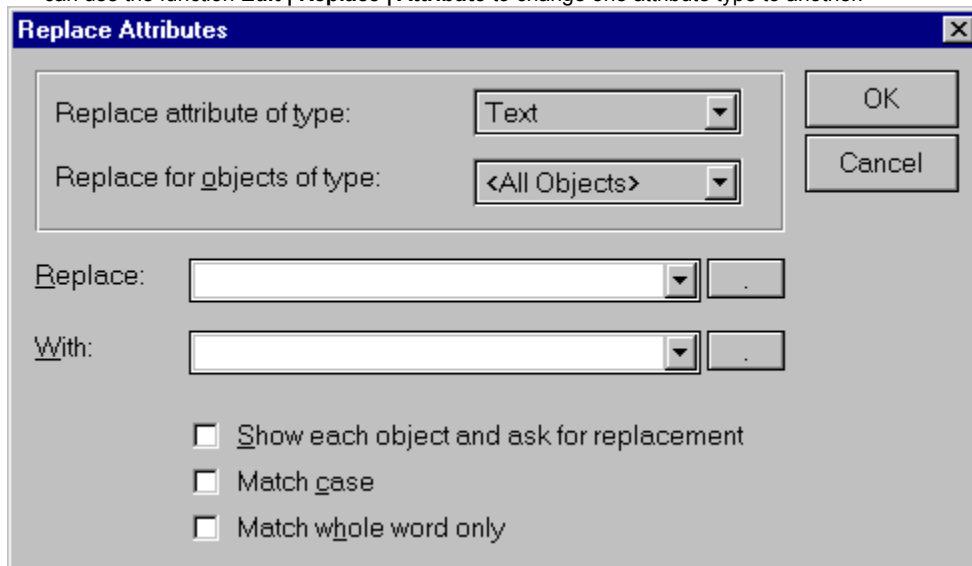
Sometimes, you may find that you need to change components; e.g., you created check boxes and you decide afterwards to change them into radio buttons. The command **Edit | Replace | Object** is used to change types of components.



Select the components to be replaced and execute **Edit | Replace | Object**. Then select the component class (*this means the type of components*) which will replace the selected components.

## Edit | Replace | Attribute

Sometimes you need to change the attributes of a range of components, e.g., you want to change the font of text fields. You can use the function **Edit | Replace | Attribute** to change one attribute type to another.



In this dialog box, you can select the type of the attribute (e.g. "Text") and for which type of components you want to perform the replacement (e.g., "All Objects" or "Push buttons" only). From the two list boxes in the dialog box, you can select the replacement.

Begin by dropping down the first list box which refers to the attribute. Select the attribute you wish to replace. Next determine which component(s) you wish the replacement to be valid for. You can apply various filters for the selection of components. The next step is to specify the actual attribute value which is to be replaced. Finally, state the value it is to be replaced with. An example would be:

"Replace the Font attribute of all selected instances of command buttons from Helvetica 14 pts to Times Roman 14 pts."

In addition, you can choose to **Show each object and ask for replacement**, to **Match a case** or to **Match a whole word only**. This facility allows you to step through the list of components and decide on a case by case basis.

## Edit | Set Bookmark

While developing your application, you might want to jump between different pages that are not immediately adjacent. You can set a bookmark on the page you wish to jump to often. This bookmark can be changed as often as you wish.

For example, let's say you set a bookmark on page 5 and you are on page 23. Execute the command [Edit | Goto Bookmark](#) and MindMap will jump to page 5. [Back to last Bookmark](#) brings you back to page 23 again. The [File | Preferences | System](#) function offers the option to show the bookmark. If you have set this option, the actual page name with the bookmark is shown in the right corner of the status bar.

## Edit | Goto Bookmark

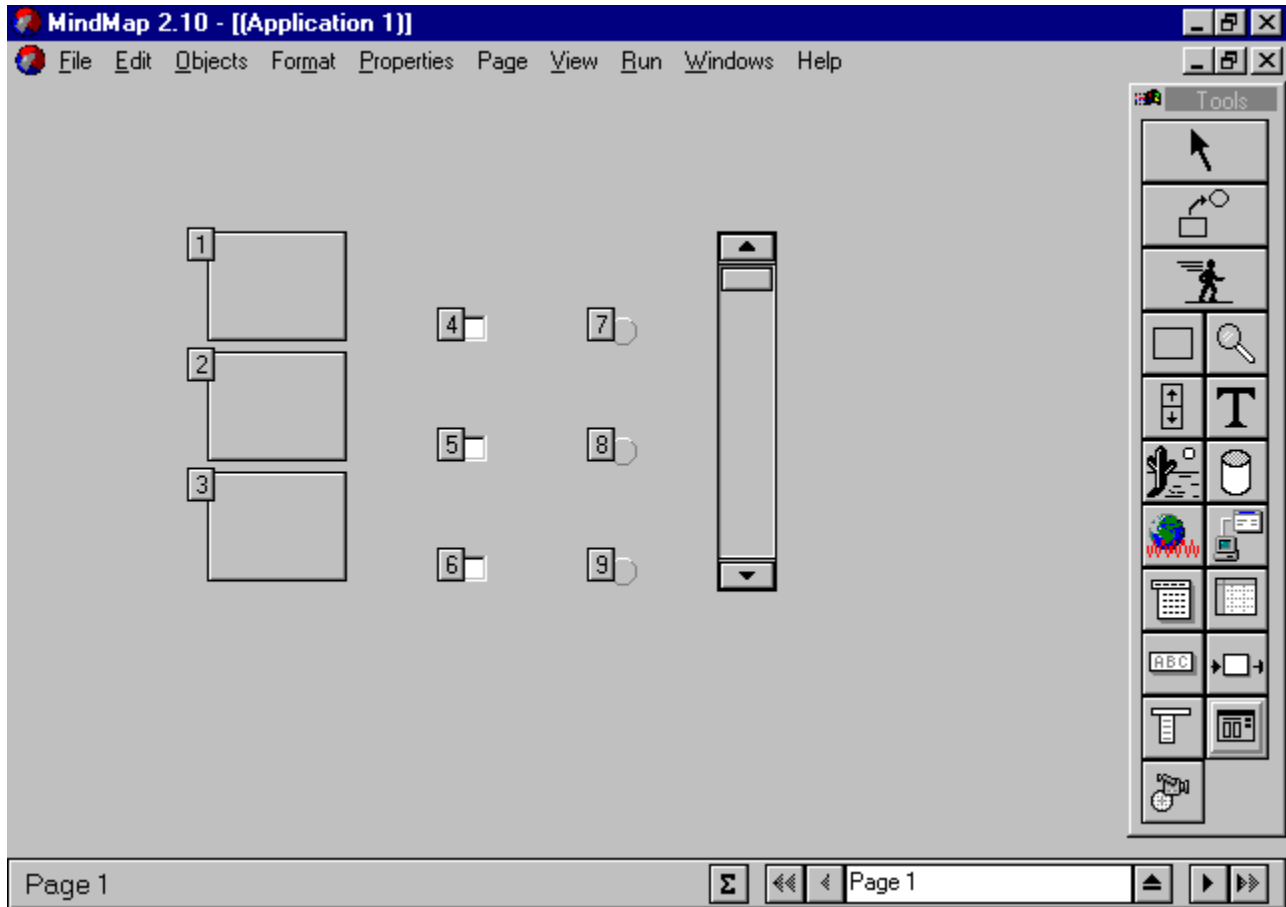
Before you can go to a bookmark, you must set it. This can be done on any page. You can quickly jump to this bookmark with the shortcut key *CTRL+W*.

## Edit | Back to last Bookmark

MindMap remembers the page from where you jumped to a bookmark and automatically saves it as **last Bookmark**. You can jump from one page to another by choosing either [Edit | Goto Bookmark](#) or **Edit | Back to last Bookmark**. You can also use the keyboard shortcuts *CTRL+W* and *CTRL+Q* for this purpose.

## Edit | Set Tab Order

Many applications require screens that have fields (*as well as buttons*) which are to be used in a defined sequence. The sequence specifies in which order each field or button is to receive the focus. The focus can be set by having the user either press the **TAB** or the **ENTER** key. The order of processing of components on the page is identical to the order of painting them. This order is initially determined by the order of their creation. If you wish to change this order to reflect some other sequence, you can either move components from foreground to background, or you can interactively change their tab order. To do this, select the menu option **Edit | Set Tab Order**.



For example, if you have a screen with some input fields, you can determine the order of their focus. With the Tab key you can allow the user to switch from one input field to the next. To do this, select the command **Edit | Set Tab Order**. You will see small buttons at the upper left corner of each component that can have focus. These buttons are numbered according to the tab order of the component. If you want to change this order, click on the components in the order you desire. The button number changes to reflect the new order. When you are finished renumbering, click on **Set Tab Order** again (*or press the ESC key*) and the small numbered buttons in the upper left corner of the components will disappear.

### Note:

You can define a selected component to be the first in the tab order by setting it to **Background** with the help of the [Edit | Background](#) function.



## Objects Menu

The objects menu allows you to choose any of the available components and place them on the screen. This method is an alternative to the selection of components from the toolbox.

### **Note:**

*Some components do not register an icon on the toolbox. A component is always required to register an entry in the Objects menu. Therefore, you will see more components in the Objects menu than are available on the toolbox.*

Since the toolbox is the easiest method of selecting the most often used components, a detailed description of those components will be offered in the section relating to the toolbox. There we will also describe the attributes of each component and the links they can have to other components. Here we will only describe actions that cannot be done with the [toolbox](#), but that must be activated exclusively in the Objects menu.

Let's begin by describing how to place components on the screen. To place any component on your screen - for example a rectangle - choose the desired type of component from the toolbox or from the **Objects menu**. The appearance of the cursor will change to reflect the selected component. It generally will be displayed as a crosshair with an attached symbol. Next, set the mouse cursor to the place on the screen where you want the upper left corner of the component to appear. Then press and hold the left mouse button, pull the rectangle open as it displays the current size and release the mouse button. The component is created and placed on the screen. If you are not satisfied with the size or location of the component, you can resize or move it as described in the sections **Selecting Objects**, **Sizing Objects** or **Dragging Objects**.

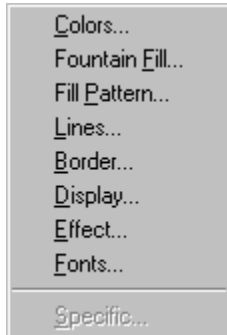
Next, let's look at how to create a new page: Since MindMap is page oriented, you can only place new components onto an existing page. You cannot create a new page by using the [toolbox](#) or the Objects menu. All actions dealing with pages must be issued by using one of the functions in the **Page Menu**, described later in this chapter, or by using the page buttons on the status bar.

MindMap offers the following components for your use:

- Draw
- Buttons
- Text
- OLE
- Printer component
- Database Manager
- WWW Browser
- Encapsulator (*Client / Server*)
- List box
- Combo box
- Data table
- Input field
- Input / Output
- Menu
- Output page
- VBX (*if any VBX control has been installed*)
- MCI

## Format Menu

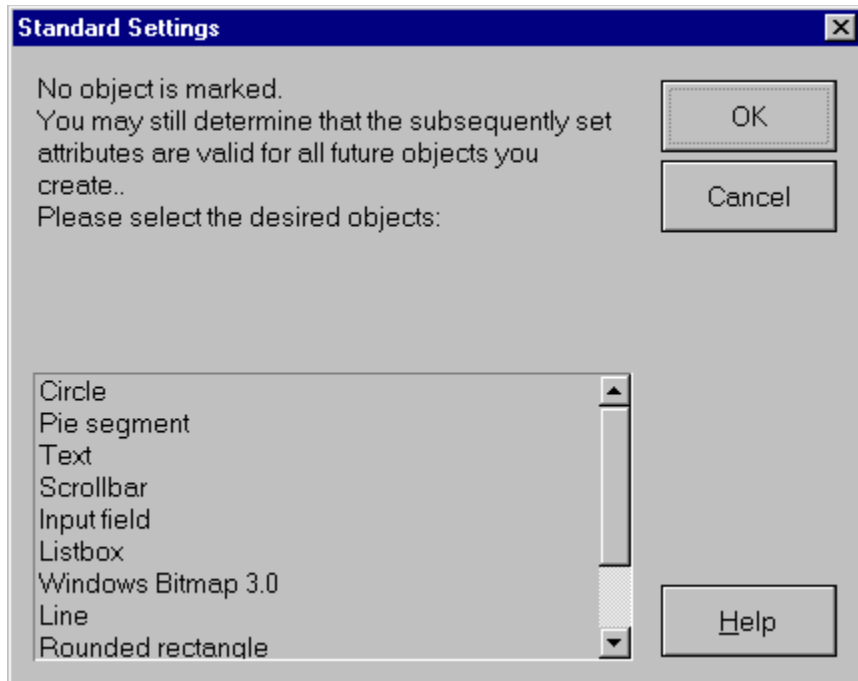
This menu allows you to change the attributes of components. Depending on whether or not a component is selected, the menu selection will either affect this specific instance of the component or all instances of a component. In other words, you can use this menu command to set default values for all subsequently placed components in this particular application.



```
{button ,JI('Ref_cont.HLP>main','Format_Border')}  Format | Border
{button ,JI('Ref_cont.HLP>main','Format_Colors')}  Format | Colors
{button ,JI('Ref_cont.HLP>main','Format_Display')}  Format | Display
{button ,JI('Ref_cont.HLP>main','Format_Effect')}   Format | Effect
{button ,JI('Ref_cont.HLP>main','Format_Fill_Pattern')}  Format | Fill Pattern
{button ,JI('Ref_cont.HLP>main','Format_Fonts')}   Format | Fonts
{button ,JI('Ref_cont.HLP>main','Format_Fountain_Fill')} Format | Fountain Fill
{button ,JI('Ref_cont.HLP>main','Format_Lines')}    Format | Lines
{button ,JI('Ref_cont.HLP>main','Format_Specific')}  Format | Specific
```

First, you can select one or more components and then make the setting changes in one or more of the options in the **Format** menu. Once you have selected one of the attributes, the appropriate dialog box displaying the available settings for the attribute will appear.

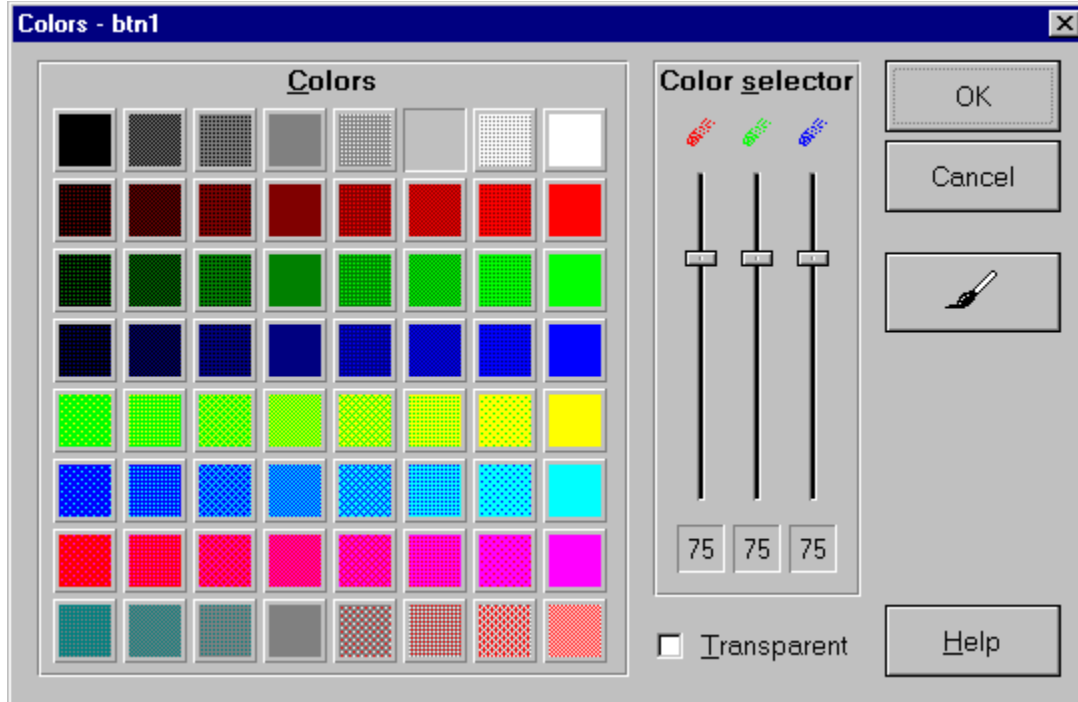
Second, you can choose one option out of the **Format** menu without having selected a component. Then MindMap will remind you in a message box that there is no component selected.



You may then choose if the subsequently set attributes are valid for all future instances of the selected type of component you create, by choosing the desired components out of a list in the message box.

## Format | Colors

This command allows you to change the color of components. Selecting this command opens a dialog box where you can choose a desired color, by clicking on the appropriate colored rectangle. The **Color selector** shows the number of the selected color as represented by the values of the three colors red, green and blue. Using the three sliders, you may modify or create a completely new color.



### Note:

If you wish a component to be transparent, check the option on the lower part of the dialog box. Thus, when you place a transparent component on top of another component, you can see the underlying component. This is quite useful in creating invisible hot spots on graphics, for example.

### Note:

You can also set the color of a bitmap to be transparent. In this case, you must also define a color of the bitmap which you would like to become transparent. This allows you to produce an effect called chroma-keying. Please refer to the section dealing with Graphics for further information.

### Note:

In order to change the color of an arrow head, you must choose the Fill Color option, as described here. Setting the Line Color of a line merely sets the color of the line itself and not the arrowhead.

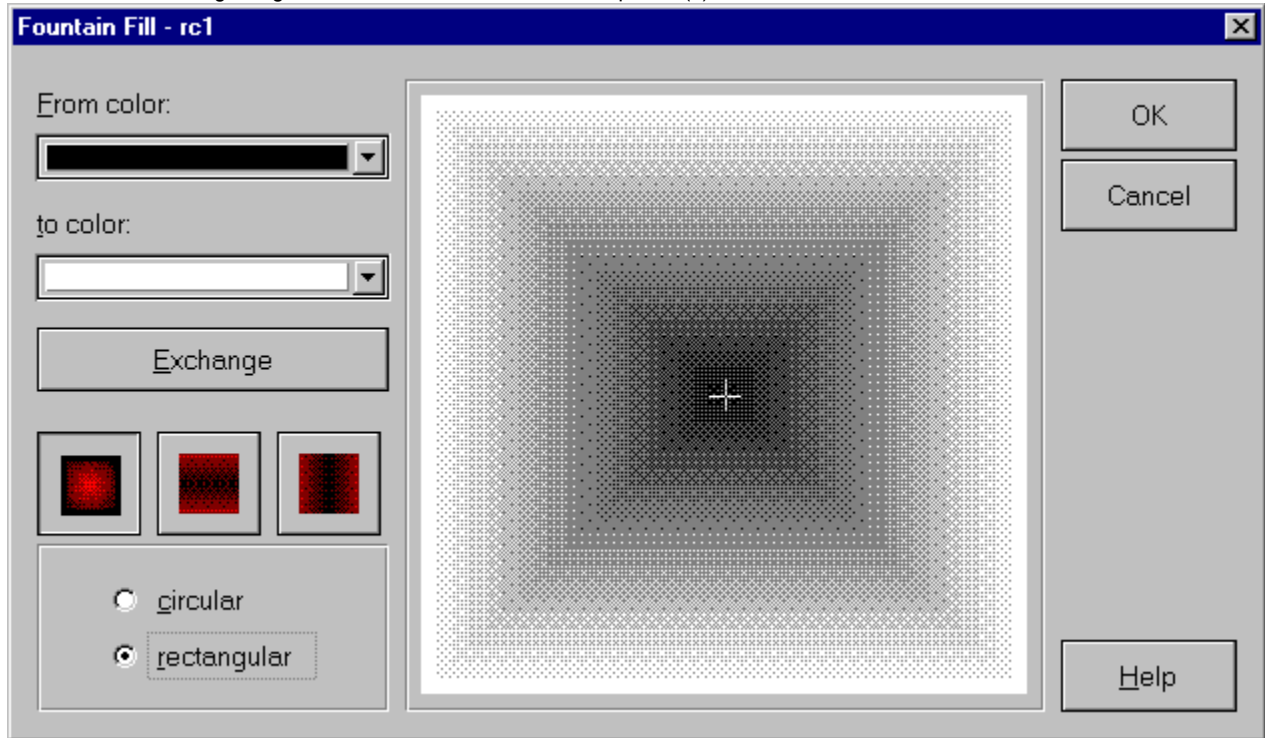
Below the OK button and the Cancel button in the color pop-up dialog box, you can see a brush. (If the selected component is capable of having a border, then the border icon will be immediately beneath the OK and Cancel button.) Clicking on that button allows you to define the color of the border of a component. This option can also be reached via Format Lines. (If the selected component is capable of dealing with the Fountain Fill and/or Fill Pattern, then these will also appear on the Attribute Toolbar.)

There are two other options underneath the brush. The first is **Fountain fill** to give draw components special effects like color toning. This option can also be reached via [Format | Fountain Fill](#). The second option is **Fill Pattern**. This option can also be reached via [Format | Fill Pattern](#).

{button ,JI('Ref\_cont.HLP>main','Format\_Menu')} **Format Menu**

## Format | Fountain Fill

**Fountain Fill** assigns a gradient color fill to the selected component(s).



Choose the first color out of the menu **From color** and define the second **color** as the **to** color. Now you can see a gradient fill ranging from the first color to the second color and which can be modified in different ways:

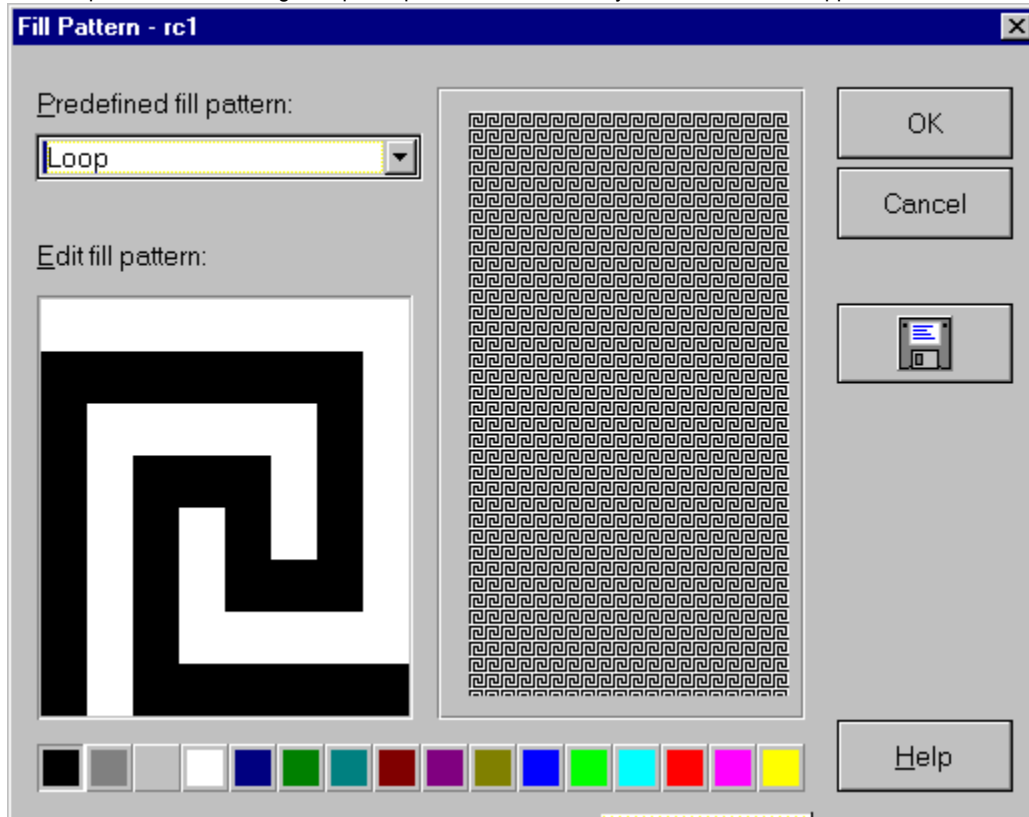
- with **Exchange**, you can reverse the colors;
- with the three command buttons, you can set the gradient fill to center, horizontal or vertical;
- and you can give the gradient fill a rectangular or a circular shape.

In the preview box, which displays the result of the current settings, you can see a little cross-hair. You can grab this cross-hair with the mouse and drag it around by keeping the left mouse button pressed. In doing so, you can determine the focal point of the gradient fill.

{button ,JI('Ref\_cont.HLP>main','Format\_Menu')} **Format Menu**

## Format | Fill Pattern

Components can be assigned special patterns, such that they have an distinctive appearance.



A selection of patterns is predefined; e.g., a loop, vertical double-lines, a small box, etc. The preview box on the right side displays the current setting. Just to the left of the display, you can view an enlarged version of the currently selected pattern. You can use this display in a manner similar to a bitmap editor. Pick up a color from one of the buttons below the editor box and begin to place the (*enlarged*) bits in the pattern. Immediately, you can see the effects on the pattern in the box on the right.

This dialog box also offers the option of loading a bitmap from the file system. Click on the button displaying a diskette and select the desired file. Once loaded, it will be visible in the editor. Again, you can choose to manipulate the bits, if you wish.

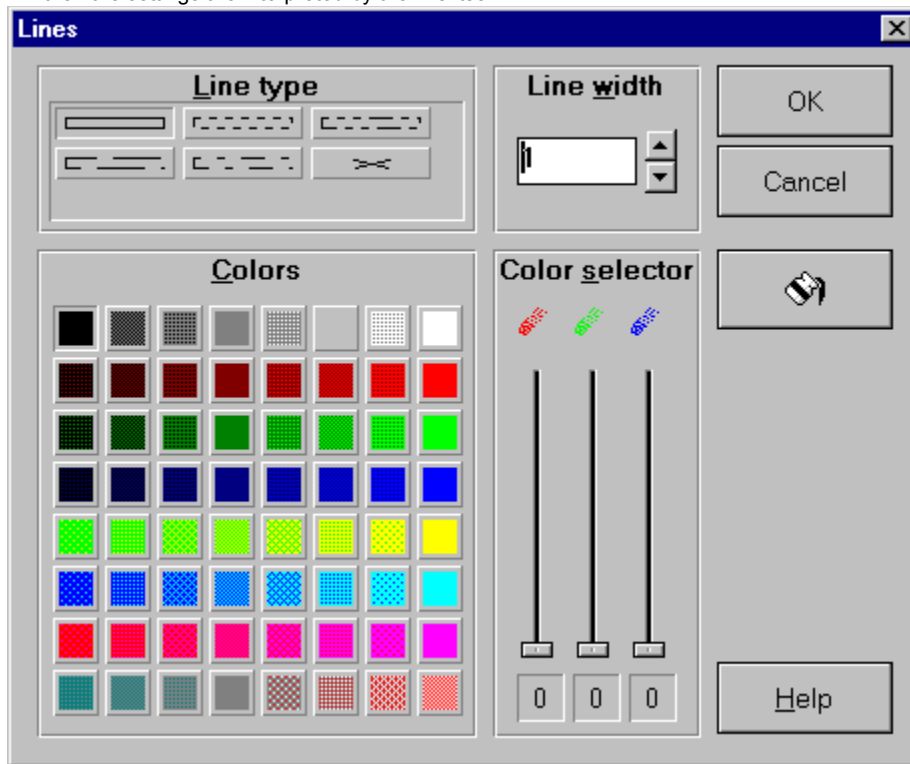
### **Note:**

*The fill pattern option only permits the usage of the 16 standard colors. If you load an external bitmap as a pattern, it will be dithered down to these colors.*

{button ,JI('Ref\_cont.HLP>main','Format\_Menu')} **Format Menu**

## Format | Lines

This menu allows you to define the appearance of the border of a component. If the selected component happens to be a line, then the settings are interpreted by the line itself.



The upper left part of the dialog box offers various settings regarding the structure of the line. Multiple options for dashed and solid structures are offered. If you wish to have a transparent line or border, select the button displaying an 'x'. Again, this is useful for creating an invisible component on the screen.

On the upper right part of the dialog box, you can set the line width. The minimum line width is 1 and the maximum width is greater than 1000.

The lower part of the dialog box allows you to set the desired color. You can either select one of the buttons displaying a color, or you can use the sliders to mix your own color.

**Note:**

*Not all colors are accepted by the line. Experimentation may be required for a color suitable in your application.*

{button ,JI('Ref\_cont.HLP>main','Format\_Menu')} **Format Menu**

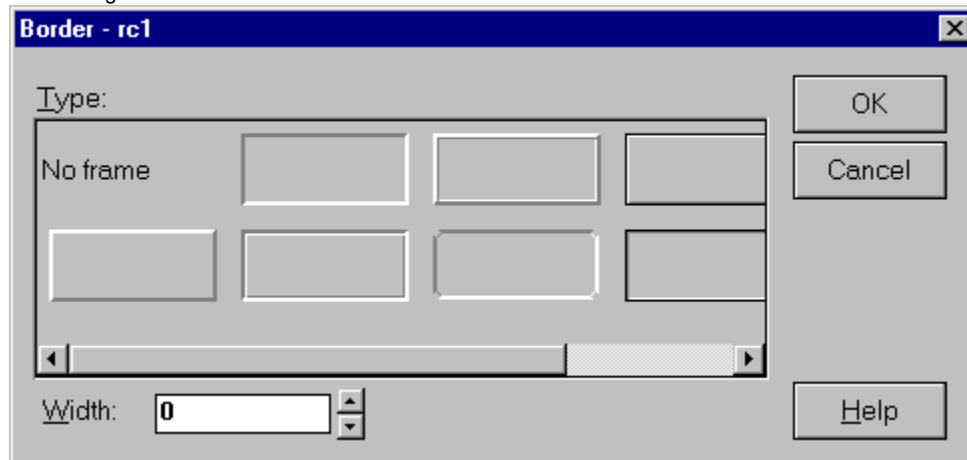
## Format | Border

You can define special borders for three component classes:

input fields;

text fields; and

rectangles.



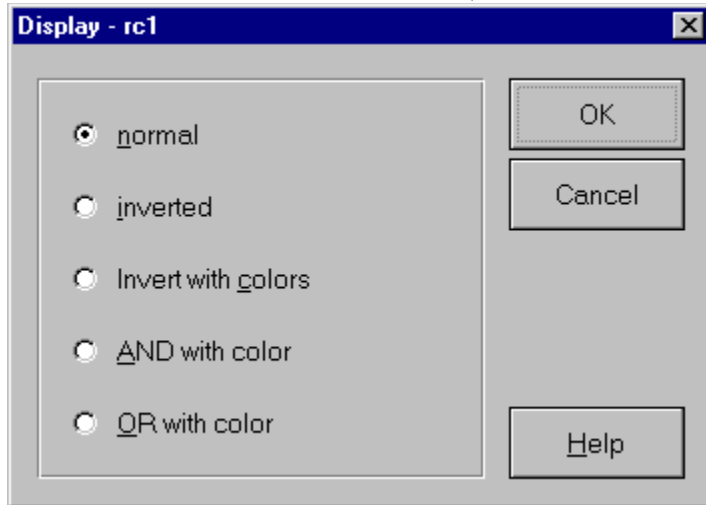
Select one of the seven predefined borders. By altering the **Width** of the border, you can further influence the appearance.

{button ,JI('Ref\_cont.HLP>main','Format\_Menu')} **Format Menu**



## Format | Display

With **Format | Display**, Text fields, Rectangles, rounded Rectangles, and Circles can be formatted with certain attributes like **inverted**, **invert with colors**, **and with color**, as well as **or with color**.



Inverted

Invert with Colors

And with Colors

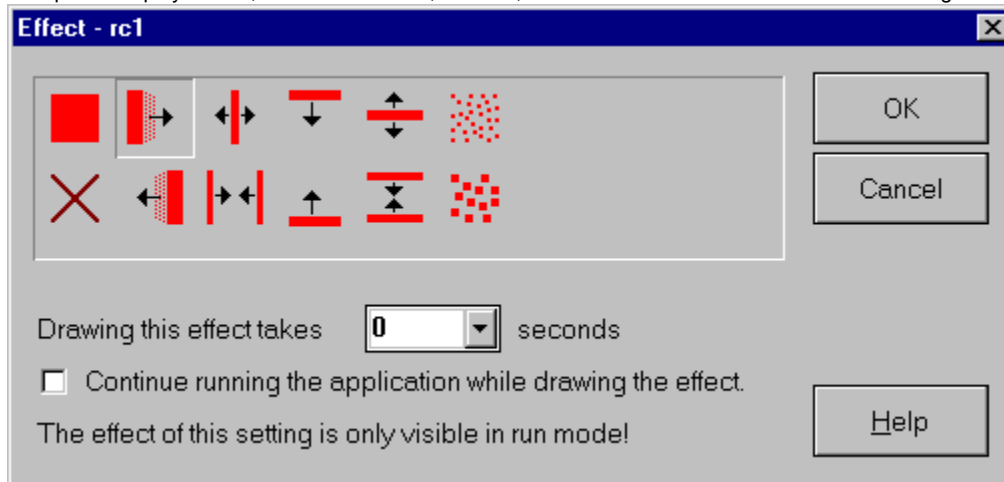
Or with Colors

These components display color and are dependent on your selection and on the color which is located behind the component.

{button ,JI('Ref\_cont.HLP>main','Format\_Menu')} **Format Menu**

## Format | Effect

You might wish to have certain components appear in a special manner during the course of your application. Often, components are used as temporary storage for values (also known as variables). These components are not intended to be displayed to the user and so you would want them to be invisible at run time. In some presentations, you might also require special display effects, such as dissolves, curtains, etc. These and more effects can be set using the **Format | Effects** option.



Let's start by describing the effects in the upper row of the **Formats | Effects** dialog box. The first (*solid*) symbol makes the component visible. This is also the default effect when a component is placed. The next symbol displays a component from left to right. The displaying of a component from the middle and out, is the third option. The fourth symbol displays the components from top to bottom. The fifth starts displaying from the middle to the top and bottom. A mosaic with small squares is built by the last option in this row. The first symbol in the second row, the X makes a component invisible. This is the most often used effect, aside from visible. The other symbols in that row should be self explanatory. To become familiar with the capabilities, try experimenting with them by assigning an effect to a component and viewing the results.

The time required to display these actions can be determined in the dialog box section, **Drawing this effect takes x seconds**. Select one of the time increments, as desired. Again, experimentation is the key to mastering the possibilities of this feature.

### Note:

*These effects are only seen in run mode.*

### Note:

*Not all components can respond to the effects attribute. If a component cannot interpret the attribute, it will simply ignore it.*

You can set an option to let MindMap continue running the application, while the effect is being drawn. If you do not set the option, MindMap will pause the application, while the effect is being drawn.

Remember that a page is also a component. Therefore, you can also assign effects to a page, so that the next page to which you are jumping, in your application, does not show up immediately, but rather with the selected effect/time combination.

Input fields, list boxes, combo boxes, data tables, and WWW browsers offer a somewhat different list of effects.



Input fields can have one of four effects: visible, hidden, keyboard entry enabled and keyboard entry disabled.

**Note:**

*If you jump to component that is hidden, the links on the component are performed and the jump returns to the "source" component to perform the next link there. If the hidden component is placed on another page, MindMap will not display this page. The links are processed anyway. This is an important feature for the use of "resource" pages. Such pages may have components that are used for various purposes, such as calculations or menu structures.*

```
{button ,Jl('Ref_cont.HLP>main','Format_Menu')}} Format Menu
```

## Format | Fonts

Various components can include text strings. Therefore, it is necessary to be able to set the font and other characteristics of the font (size, weight, color, etc.) for such components.

Select the desired component, click on **Format | Fonts** and the font dialog box will open. Select the desired font. The fonts that are available depend on which fonts are installed on your system.

**Note:**

*MindMap does not support font embedding. If you plan on running your application on other PCs, it is suggested you use a standard MS-Windows font. If you need a non-standard font, you must ensure that it is on the target systems, or take care to distribute it along with your application. In the later case, you must also ensure it's proper installation, as well as deal with any associated license issues.*

{button ,JI('Ref\_cont.HLP>main','Format\_Menu')} **Format Menu**

## Format | Specific

Each component is permitted to register its own private attributes, if it has any. These are specific to the component and thus are not shared with any other component. All specific attributes are accessible on the menu **Format | Specific** or through the icon on the attribute toolbox for a given component.



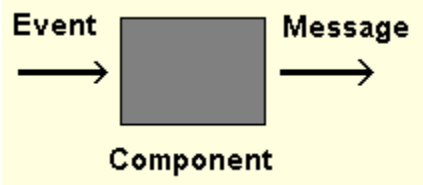
The contents of these specific dialog boxes are dealt with on an individual basis. You can find the necessary information by: 1) looking up the component elsewhere in this manual; 2) addendums provided with new MindMap-provided components; or 3) with documentation provided with any third party components that you acquire.

{button ,Jl('Ref\_cont.HLP>main','Format\_Menu')} **Format Menu**

## Properties | Links

This feature allows you to access the dialog box to define links for components. This determines the actual behavior of your application. You can access this feature by clicking on the icon on the attribute toolbox, via the menu option, or by pressing the function key *F6*.

Any component in MindMap, including a page itself, can have links assigned to it.



A link is the combination of an incoming event and an outgoing message. You cannot define an event without associating a message. This prevents you from defining syntactically invalid statements.

### **Note:**

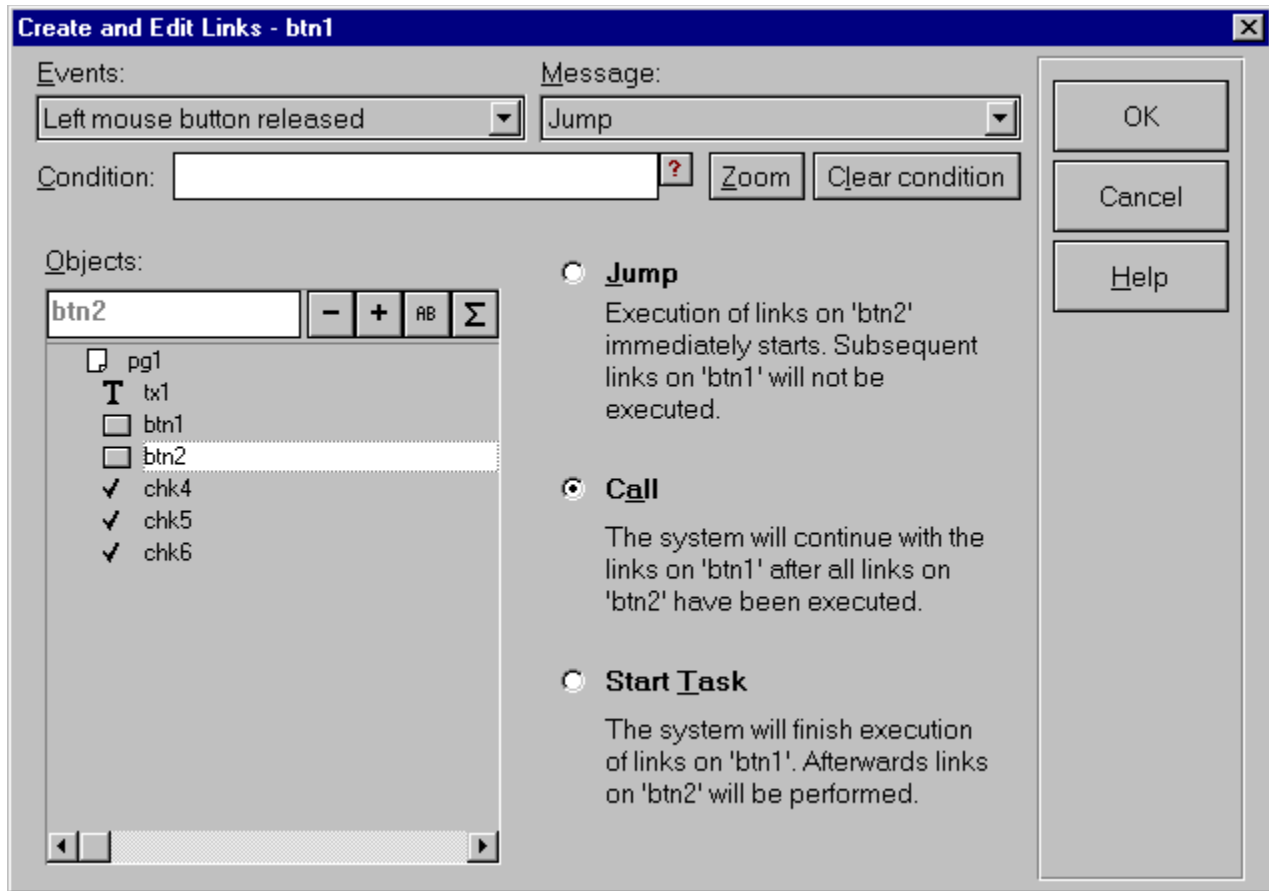
*In order for a message to be executed, an event must occur. MindMap is a truly event-driven environment. Traditional programmers, who tend to think in terms of flow and procedures, must rethink the paradigm and consider development as defining a string of events and messages.*

Most components share a common set of events and messages. A component can also register specific events, which only make sense for that specific component. The same may be done in terms of messages. A component can register a specific message, which becomes available to all other components.

To define the behavior of a component, first define the event on which it is to trigger. Then select the message that it will generate, contingent on the occurrence of the event. You can also associate a condition with a link, thus limiting the execution of the message to certain situations or conditions.

A few examples are:

- When the left mouse button is released, change the fill color of a rectangle.
- When the mouse is moved on top of a check box, display a message on the status bar.
- When the left mouse button is released, jump to page number five, if the amount is greater than 500.

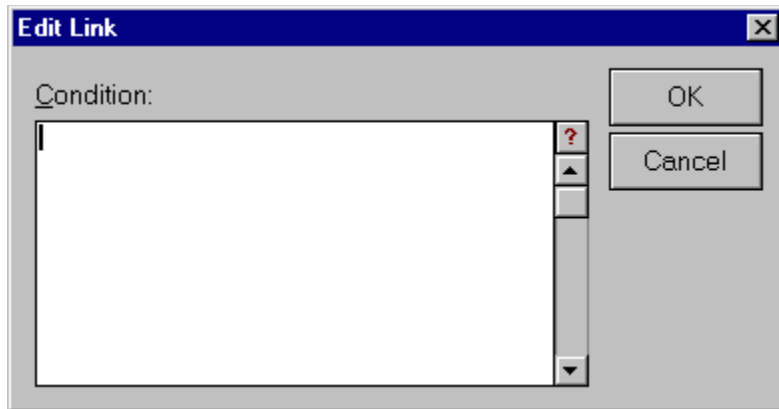


An example of an event is "Left mouse button released". An example of a message is "Jump to Page 2". Let's assume you have an application consisting of two pages, with a command button and an input field on page one. The command button contains the link "When left mouse button released - jump to page 2". In run mode, if you click once with the left mouse button when the mouse cursor is over the command button, you will see the command button move down and up. The event "Left mouse button released" is then handed over from MS-Windows to MindMap. This causes MindMap to jump to page two ("Page 2"), which means that page one ("Page 1") disappears and page two will be shown on the screen.

In addition, you can set a **condition**. In the example above, you see that the link will only be performed when the condition "edt1 = 1" is true. Let's explain this. On page 1 we have placed two components - one command button ("btn1") and one input field ("edt1"). The message "Jump" will be performed when the event "Left mouse button released" is triggered by the component, but only if the condition "edt1 = 1" is true. The condition is true if someone has input a "1" into the input field or if another link has assigned the value "1" to "edt1".

In the upper half of the **Create and edit links** dialog box, you see a command button labeled **Without condition**. By clicking on it, you can delete the condition formula, if one is in place.

To the right of the condition dialog box is a button labeled **Zoom**. When you click on it, an additional dialog box pops up:

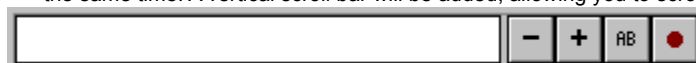


If your condition statement becomes large, and requires more space to view than is available in the small field of the **Create and edit links** dialog box, you can use this larger dialog box to edit your condition. A scroll bar on the right side allows you to edit even longer conditions than would otherwise fit into this field.

By clicking on the button labeled with the question mark, you will get a list of all parser functions that are built into MindMap, along with a short description. To learn more about functions: *see the section about the [Parser](#).*

Also defined in the **Objects** dialog box is to which component the **Jump** will lead. Our list of components shows four components, with the name of component "Page 2" highlighted. Highlighting a component results in displaying its name in the text field in this dialog box.

When you have large applications with numerous components, the list box will not be large enough to show all components at the same time. A vertical scroll bar will be added, allowing you to scroll through a list of all the components in your application.



When you open the **Create and edit links** dialog box of a component, the components in the Objects dialog box will be ordered by pages and within the pages by their creation order. By clicking on the "-" sign, it collapses the list to only display pages. That allows you to quickly find a desired page. By clicking on the "+" sign, the list will be expanded again. If a page has components placed on it, these will be displayed in the order of their creation.

Clicking on the "AB" button sorts the components in alphabetical order. For example, you need not scroll through all of your component names to reach one that starts with the letter "Z". Place the mouse cursor into the white field on the left of the buttons by clicking on that field and key in a "Z". The display of the list of components will immediately change and start with components whose names begin with a "Z", if any exist.

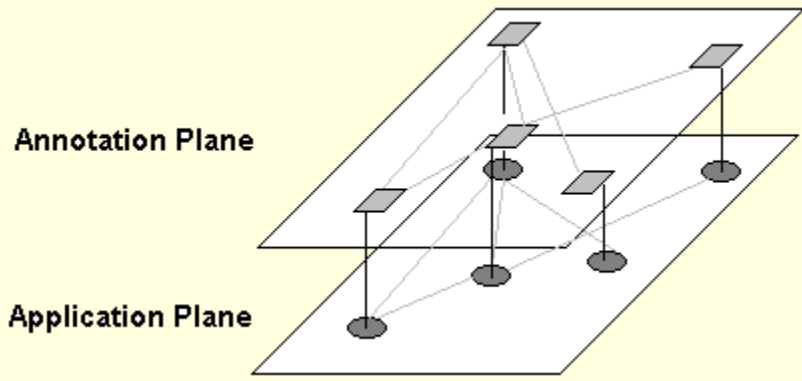


## Properties | Note

MindMap offers a feature to help you produce system documentation for your application. This feature is available in the menu **Properties | Note**, or via the **Attributes toolbox** that opens by clicking on the component. Here you can type in your annotations (*comments, for those of you familiar with coding*) and refer to them later.



The idea behind the annotation feature is the goal to physically separate notes or comments from the 'code' in an application, without losing the logical connection. Annotations are tagged onto components. Since the relationship between components is well-defined, via the links structure, the annotations can be mapped on top of the logical structure and connected by means of hyperlinks. This feature allows you to view comments without having to progress through the 'code' to find them. In fact, you may navigate through the entire application by this means. This is especially useful in very large, complex applications.



You can print the annotations of one or more components by using the function **File | Print | Objects and annotations**.

When you have selected a component and open the annotations dialog box, you can look at the annotations of other components associated with it by "jumping" to those components. For this purpose, you must select one of two lists of components by double-clicking on the text **From this object** or **To this object**, in the **See also** section of the dialog box. All components that have links to or from the selected component, will be listed after this double-click. If desired, you can set a check mark to show only those components which have annotations. After highlighting a component, you can click on the **Go to object** button. MindMap will deselect the current component, select the highlighted component, and show the annotation of the newly selected component in the **Description** text field. You can easily jump back to the previously selected component by clicking on the **Back** button.

## Properties | Drag & Drop Options

The **Drag & Drop** option works with input fields, data tables or list boxes, for example. It allows you to select the option for this component to send or to receive data via drag & drop, by setting corresponding check marks in the dialog box.

This option is also available for some components, via an icon in the **Attributes toolbox**.

Setting the **Drag&Drop** option is only necessary if you want the user to perform **Drag&Drop** operations with the mouse.

Where **Drag&Drop** operations are performed by links (*events*), they can be used without setting the **Drag&Drop** options, because what is dragged and dropped is under control of the application.

For example, when you place two input fields on a MindMap screen and you select Drag&Drop **send data** for the first one and **receive data** for the second one, then you can perform the following operation in run mode: Type in some letters or a number in the first input field. Next, hold the left mouse button down while the cursor stays over the first input field. Then drag the cursor over the second input field and release the mouse button. You will see the content of the first input field is copied into the second input field.

You can use this feature for copying the contents of input fields, data tables and other components to one another. By choosing the appropriate settings, you can even drag the contents of components onto an output page and let the user create his own output in run mode.

Input fields, list boxes, combo boxes, output page components and data tables have additional setting options for **Drag&Drop**. For more information, see the description of these components elsewhere in this manual.

## Properties | Runmode Options

MindMap offers some functions that are only visible in **run mode**.



If you select a bitmap, or one of the graphical primitives for example, you will get the following run mode options in the **Properties Menu**: Move, Size, Options.

- **Move** options are only available for graphic primitives and bitmap components. By default, the **Move** option has check boxes for **no**, **any**, **horizontal only**, **vertical only** and **along the line**. (In the later choice, you can select a line that you have previously placed on the screen.)
- **No** will not allow a component to be moved around with the mouse on the screen. This is the default setting.
- **Any** gives the option to move a component to any position on the screen.
- **Horizontal only** allows the user to move a component only horizontally, starting from it's original position.
- **Vertical only** allows the user to move a component only vertically, starting from it's original position.
- **Along the line** allows the user to move a component along a line component, which you have previously placed on the screen. You can select the desired line from the list below the text **Along the line**. If you have selected this option, you can select the additional options **Move in anchor points** or **Drag in anchor points**. This will restrict the movement of the selected component to the anchor points (*nodes*) of the line. When you have not selected **Along the line**, these two options are not available and are grayed out.

If you have chosen anything else but "no", you can set the option **Create instance**. (*If you have selected "no", this option is grayed out*). With this option, the user of your application can generate a "copy" of the component. It can be moved to another location on the screen in run mode, as is allowed by the settings in the move options. The components that have been generated by this method in run mode will disappear when you switch to edit mode.

The **Size** options allow you to change the size of the component when it is moved. With the **Set** button you can assign the current size of the selected component to the **min. horizontal** and **min. vertical** fields. If you do not want to allow a change in size, check the option **fixed**.

The **Options** selection allows you to choose which component in your application will be allowed to move. Select the desired component from the list that you can open by clicking on the button on the right side of the list box.

## Properties | Arrange

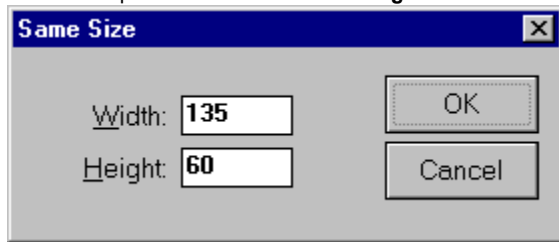
If you have components on a page, you can arrange them, relative to each other, in six different ways on the screen.



First, you must select the components you wish to arrange. You can then align them horizontally or vertically. The symbols in the **Arrange** option are described from top to bottom as follows: The first two options arrange the components in a virtual line that goes through the middle of the components. You can align the components vertically left or right, i.e., the virtual line where the components line up is at the left or the right side of the components. The last options allow a horizontal alignment of the components. In this case, the components orient themselves by their upper or lower border.

## Properties | Same Size

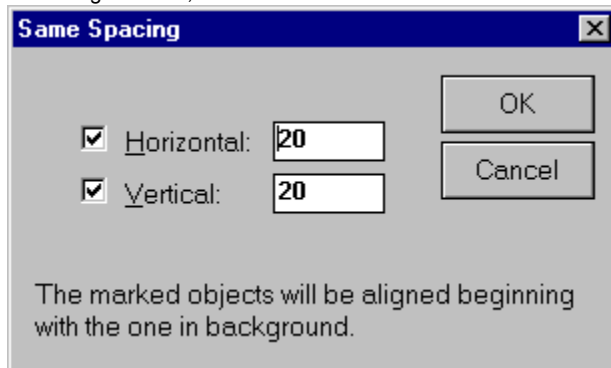
You can create components of the same size by copying or duplicating them from one component that has the desired size. In cases where you have already created components in different sizes, but which you later want to be a common size, MindMap offers a convenient **resizing** feature.



To resize one or more components, select the desired components and click on **same size**. You will get a dialog box with input fields for the common **width** and **height** of the selected components. By default, these numbers show the size of the component that has the lowest tab position (farthest in the background). You can override these numbers by typing in the desired size of your components. The size is given in pixels.

## Properties | Same Distance

With this function, you are able to define the **spacing** between the components you have selected. You need to state the pixel size and whether the distance should be horizontal, vertical or both. You must blank the appropriate field to avoid setting the wrong distance, if it is not desired.



Same Spacing

☒ Horizontal: 20

☒ Vertical: 20

OK

Cancel

The marked objects will be aligned beginning with the one in background.

The selected components will be aligned beginning with the one farthest in the background. Do not confuse this with the apparent visual position. Whether or not a component is 'behind' another one depends on its time of creation, not its position on the screen. To view the relative position of a component, switch on the [Edit | Set Tab Order](#) option.

## Properties | Group

If an application has some components that belong together, or if they should have a common link (event), you can group them. That means that you can move them, for example, without changing their internal position or you can use these groups as if they were one single component in the Links dialog.

If you are using **Radio buttons**, you can use the **Group** function to determine which buttons will act as one unit. Imagine two groups of three **Radio buttons** that are arranged on one page. Clicking on one button within one group will not affect the settings of the buttons of the other group, but will reset the other members of its own group.

You can still access the **object specific attributes** of a single component that is a member of a group by clicking on the group. A pop-up window lists the component members of the group, so that you can get the attribute toolbox of a specific component after selecting the component from the list.

## Properties | Ungroup

This command is available for components that are already grouped. After having selected a group, you can ungroup the components of that group. After ungrouping the components, the group itself is no longer available, as it no longer exists. Because any group can be the target of [Links](#), you will be asked by MindMap if you really want to ungroup the components, if such Links do exist.



## Properties | Fix

Components that never or seldom change their position, or their attributes, can be fixed so that they are always in the same state. To fix a component, select it and click on **Properties | Fix**.

## Properties | Unfix

This command unfixes components that have been previously fixed. As you cannot select a fixed component, unfix effects all fixed components on that page.

## **Page | New Page**

This command adds a new page at the end of an application and makes it the active page.

## **Page | Insert Page**

This command inserts a new page in front of the active page; i.e., if your application has three pages and page two is active, this command will create page number four between page one and page two.

## **Page | Delete Page**

If a page is no longer needed, you can delete it with this command. This command also deletes all components on that page without warning. If the components contain links or are the target of links, you will be asked to confirm the delete function. You will not be asked as long as the components on the page do not have links themselves or are not targets of links.

## Page | Next Page

You can move to the next page of your application with this command or by using the shortcut function key *F8*.

Another way to move to the next page of your application is to click on the button with the right arrow in the status bar. With this button, you can also create a new page when no further page exists.

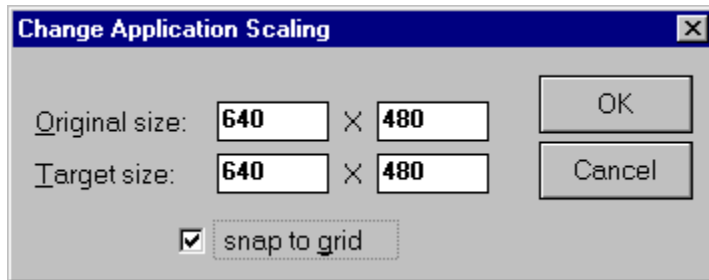
## Page | Previous Page

You can move to the previous page with this command or by using the shortcut function key *F7*.

Another way to move to the previous page of your application is to click on the button with the left arrow in the status bar. It's the one located to the left of the field which shows the name of the selected component.

## Page | Scaling

This option allows you to change the scaling of the application. MindMap shows the original size of the application and asks for the target size (in pixels) in a dialog box. You can switch an application that was developed in 640x480 pixels to 800x600 and vice versa with this feature.



By default, the components will be scaled proportionally. This can lead to errors due to rounding inconsistencies. The option **Snap to grid** can be used to adjust the new component size to the actual grid setting in the [File | Preferences | Screen](#) section. You may find it advantageous to deselect this option, depending on the types of components you have in your application. In particular, care should be exercised when scaling down, as fonts tend to be problematical.

**Note:**

*You can also use this option to have the user set the appropriate screen size at runtime. Include a selection button at the outset of your application which calls this menu option.*



## View | Full Screen

It is sometimes useful to have a complete view of an application page. For example, if you are developing an application in a resolution of 640x480 pixels and your monitor has the same resolution, you will not see the whole page. This is due to the overlapping status and menu bars. The full screen function or the shortcut function key *F9* changes the appearance of the MindMap screen in such a manner that the Menu and status bar are hidden. You can now see more detail on your screen.

## View | View Toolbox

You can hide or show the **toolbox** with this command or by using the shortcut function key *F10*. We suggest you use **View Toolbox** as the default.

## **View | View System Log**

MindMap creates its own system log, allowing you to see each action that MindMap is taking behind the scenes. It is a visual representation, in a dialog box, of the file, MMERROR.LOG, that MindMap has written. The information that can be viewed here can be helpful in the development process of an application because you can see which error message the parser reports, which database operations are executed, etc.

## View | View Status bar

You can hide or show the status bar with this command. We suggest you retain **View Status bar** as the default.

## View | View Guidelines

Guidelines are helpful as you are designing a complex screen layout requiring precise orientation of components. To hide them, use this command. See how to set guidelines in the next section.

Guidelines are helpful as you are designing a complex screen layout requiring precise orientation of components. Once you have selected **Set Guidelines**, you are given the choice of either horizontal or vertical lines. The cursor now changes to a double headed pointer and by clicking the left mouse button, you will create guidelines until you select the pointer in the toolbox to turn this off. The exact position of a selected guideline is shown in the status bar.

Guidelines cannot be selected and so the act of deleting guidelines is different from the deletion of other components. Position the cursor on a guideline, press the left mouse-button and then drag the vertical guideline off the left or right side of the screen. The same procedure applies to horizontal guidelines except that you drag them to the top or bottom of the screen.

## View | Page Overview

In most cases, a MindMap application has more than one page. To see the pages as small thumbnails in an overview or 'slide show' representation, choose **View | Page Overview** or use the shortcut function key *F11*. You will see a small representation of the pages and the page names. With a double-click on a specific page, you can easily jump to this page.

## View | Zoom

MindMap offers a magnifying glass feature so that you can **Zoom in**, **Zoom out** or switch to the **Normal** view. You can zoom in and out several times depending on the degree of magnification. The feature is available via the menu command **View | Zoom** as well as an icon on the toolbox.



Once you have chosen **View | Zoom** the cursor will change to a magnifying glass. Press the mouse button and drag the resulting rectangle over the area you wish to magnify. To zoom out again, click on **Zoom out** or **Normal**.

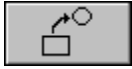
## Run | Calculate

This function forces a new calculation of all formulas in the components of your application. You can initiate the recalculation either via the menu command **Run | Calculate** or by pressing the shortcut function key *F12*. Note that MindMap executes these calculations in run mode whenever an event is executed, such that in most cases you do not have to use this function. It is important to note that an entry into an input field in a MindMap application will not automatically perform a recalculation. Only if an event has been defined, will keying in data cause a recalculation. More technically speaking, the parser will not begin to process variables until it has received control. It will not receive control until the event-message engine has been launched.



## Run | Link

The links of a selected component can be viewed graphically, if you choose this option or use the shortcut function key *F3*. A pop up window appears in the lower right corner of the screen and shows all the links of a selected component. A click on another component shows its links. (*see also the description of this option in the [toolbox](#) section*)



If you are looking for specific links in your application, the link option can also be used. Select one of the **Events** in the **Link** dialog box and MindMap will show all components that have this link with a special icon that symbolizes the event. Now you can navigate through the pages of your application and you will see all the components with this selected event.

By dragging the link dialog box away from its original position (*tearing it off*), it will stay on top of your screen until closed.

By clicking on the pointer on the [toolbox](#), the Link dialog box will close.

## Run | Run Mode

The shortcut function key *F4* also toggles you between run mode and edit mode. By switching to run mode, you can work with your application immediately, as there is no compiling in MindMap.



You can also switch to run mode by clicking on the corresponding button in the [toolbox](#).

**Note:**

*To exit the run mode, you can double-click on the extreme right edge of the MindMap screen. This shortcut will only work if the application does not display vertical scroll bars. This may not work if you are running the application in a window which does not fill the screen, but you can always use the *F4* key to switch back to edit mode at any time.*

## Run | Breakpoints

During the development of your application, you might discover errors or unwanted courses of events. In order to assist you in locating these errors, MindMap offers a **debug mode** so that you can step through your application one link at a time.

Depending on the type of error in the application, you can set a breakpoint on an **Event** or on a **Message**. Select the appropriate feature out of the **Breakpoints** pull-down list and switch to run mode. The application will pause during run mode as soon as the first of the selected breakpoints is encountered. Now you can go through your application step by step and observe what happens. If you choose **Step**, MindMap will carry out the next link and pause at each subsequent link. If you choose **Run**, the program will resume and pause only at the next breakpoint.

It is quite common to have events that you wish to occur only if a certain condition is either true or false. You can define these conditions in the **Links** dialog boxes of the components. In **debug mode**, you can see the condition of the "if"-button. At the same time, you will see a "1" or a "0" in the evaluation button. "1" signals that the condition is true and "0" that it is false. If you do not see either "1" or "0", you have an incorrect statement in the condition, such that MindMap is not able to interpret the condition. This allows you to test if your conditions have been written in the right syntax and if your application branches to the desired links depending on the conditions.

The **Run | Breakpoints** feature is very useful in determining the order Links are executed by MindMap and if the application behaves as you want. In order to guarantee that you can trigger on the very first events, be sure to choose **Application started** as the first event.

The **Run | Breakpoints** feature offers you various options for debugging your applications by using the five command buttons at the bottom of the dialog box:

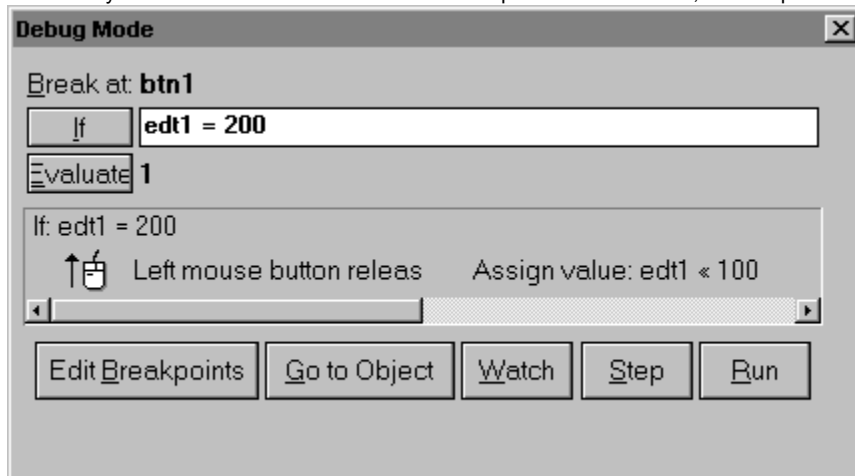
With a click on the button **Breakpoint on event**, a list of all events the component can recognize is displayed. You can choose one event out of the list. In run mode, the application will pause when this event is executed the first time. You can then evaluate the displayed information and proceed by clicking the **step** button or the **run** button. Notice that you can choose more than one event to break on.

In addition to choosing an event, you can also choose a message to set a breakpoint on. Just click on the **Breakpoint on message** button.

With a click on the button **Delete Breakpoint**, you can deselect an event that is highlighted in the list box, showing the previously chosen events.

How to use the Breakpoints feature in run mode.

When you switch to run mode and the first breakpoint is encountered, MindMap will display a dialog box:



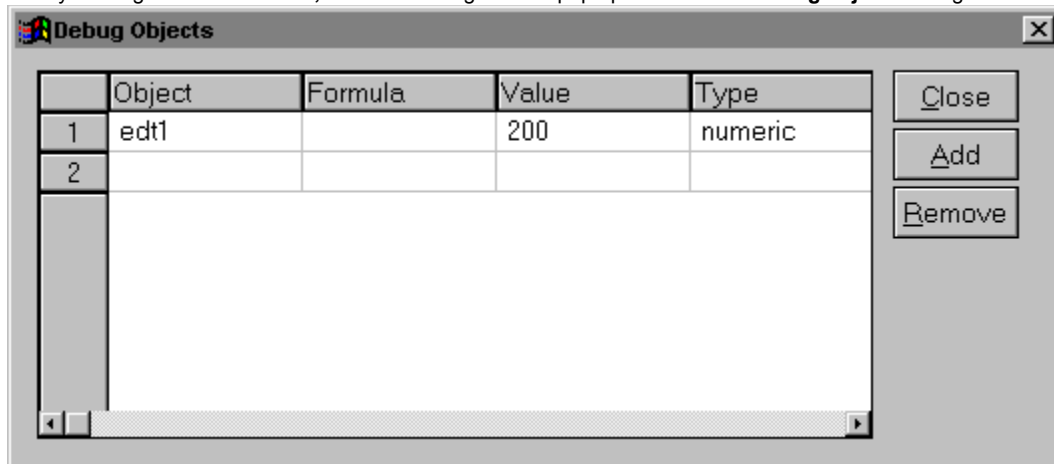
By clicking on the button **Edit Breakpoints**, MindMap will show a dialog box where you can make all basic breakpoint selections, as described above.

By clicking the button **Go to Object**, you will be asked if you want to terminate the run mode. If you answer with "Yes," MindMap switches to edit mode and will select the current component.

A double-click anywhere in the field where the link is displayed, opens the **create and edit** dialog box. There you can edit the

link. Your changes will be permanently saved with the component.

By clicking the button **Watch**, a second dialog box will pop up. It shows the **Debug objects** dialog box.



Here you can evaluate components, their formulas, values and types. To include a component, click on the **Add** button. This will open a dialog box that lists all components of the application. Choose the desired components by clicking on them and they will appear with their additional information in the **Debug Objects** dialog box. If you know the component name, you can simply begin typing it. If you click, for example, on the **run** button of the **Debug Mode** dialog box, you can watch how values change, depending on your application.

To remove a component in this mode, highlight a component in the **Debug Objects** dialog box by clicking on the corresponding number in the left most column of the dialog box and then clicking on the **Remove** button. The component will be removed from the list, not the application.

You can close this dialog box with a click on the **Close** button.

Note that the **Close** button of the Breakpoints message box will only close the display dialog box, but will not terminate the breakpoint debugging mode.

**Note:**

*Another method of calling the Debug dialog box is by pressing the SHIFT+F4 keys*

## **Windows Menu**

This is a standard MS-Windows menu command. It arranges the windows on the screen and switches between different applications. Please review your MS-Windows documentation for more details on this topic. If you have two or more MindMap applications open at the same time, you can switch between these applications at any time, by utilizing this menu.

## Help Menu

If you click on **Index**, MindMap switches to the help index. If you click on **Keywords** the program switches to an alphabetical keyword list. Please review your MS-Windows documentation for more details on using this help system. It works in the same manner as any other standard MS-Windows help system.

The **About MindMap** entry produces the following screen:



In the bottom left corner of this screen, you will find the first 10 digits of the serial number assigned to your copy of the software. This number is your key to various MindMap services, such as online support. The actual serial number consists of 14 characters, of which the last four are considered your personal identification number (*PIN*). These are not published and should not be revealed to the public.

## The use of the Toolbox

Within this element, you have most of the important and often used components that are needed for developing MindMap applications. We therefore advise you to use **View Toolbox** as the default in the **View** menu, so that you always have quick access to the different components in the toolbox.



The toolbox itself can have more than ten components, depending on the modules you are using with the application development environment.

The actual sequence of icons on the toolbox is dependent on the order in which the components are loaded. Normally, the user will not influence this order, but MindMap does offer an option whereby components can be explicitly loaded or, on the other hand, kept from loading. This is controlled by an entry in the MINDMAP.INI file. Under the heading:

[Libraries]

Default = \*.MDL

This will cause every component library (*MDL = MindMap Dynamic Link Library*), that is located in the home directory, to be loaded. This can be altered, though. Actually, you can even create your own personal load scenarios. To do this, locate the [Libraries] entry in the MINDMAP.INI file. Then enter the desired files, according to the following scheme:

[Libraries]

Section1 = DEMO1

[DEMO1]

Default =

Lib1 = MMEDIT.MDL

Lib2 = MMDATA.MDL

[DEMO2]

Default =

Lib1 = MMBASE.MDL

Lib2 = MMODBC.MDL

Lib3 = MMEDIT.MDL

Lib4 = MMDATA.MDL

In this example, only the components contained in the two libraries (*MMEDIT* and *MMDATA*) will be loaded. If you later wish to load a different set, then you would merely change the referenced section and the heading [Libraries].

Let us now specify the standard components in the toolbox. Beginning at the top they are as follows:

- Pointer
- Link
- Run mode
- Draw
- Zoom
- Button
- Text
- Graphic import
- WWW Browser
- Encapsulator (*Client / Server*)
- Database
- List box
- Data table
- Input field
- Input/output
- Menu
- MCI
- VBX (*if a VBX control has been installed*).

These components are described in section [\*Component Types\*](#).

The toolbox also contains switches which affect the option settings of MindMap:



## Pointer Option

When you start MindMap you will see an arrow on the screen. This arrow is your standard mouse pointer in edit mode. You can use the pointer to select components on the screen, pick components off the toolbox, make menu selections and execute other MindMap functions. Whenever the cursor changes from the arrow to another shape - because you selected one of the components - you can switch to the pointer again by clicking on the upper arrow in the toolbox or by pressing *CTRL+Y* simultaneously.

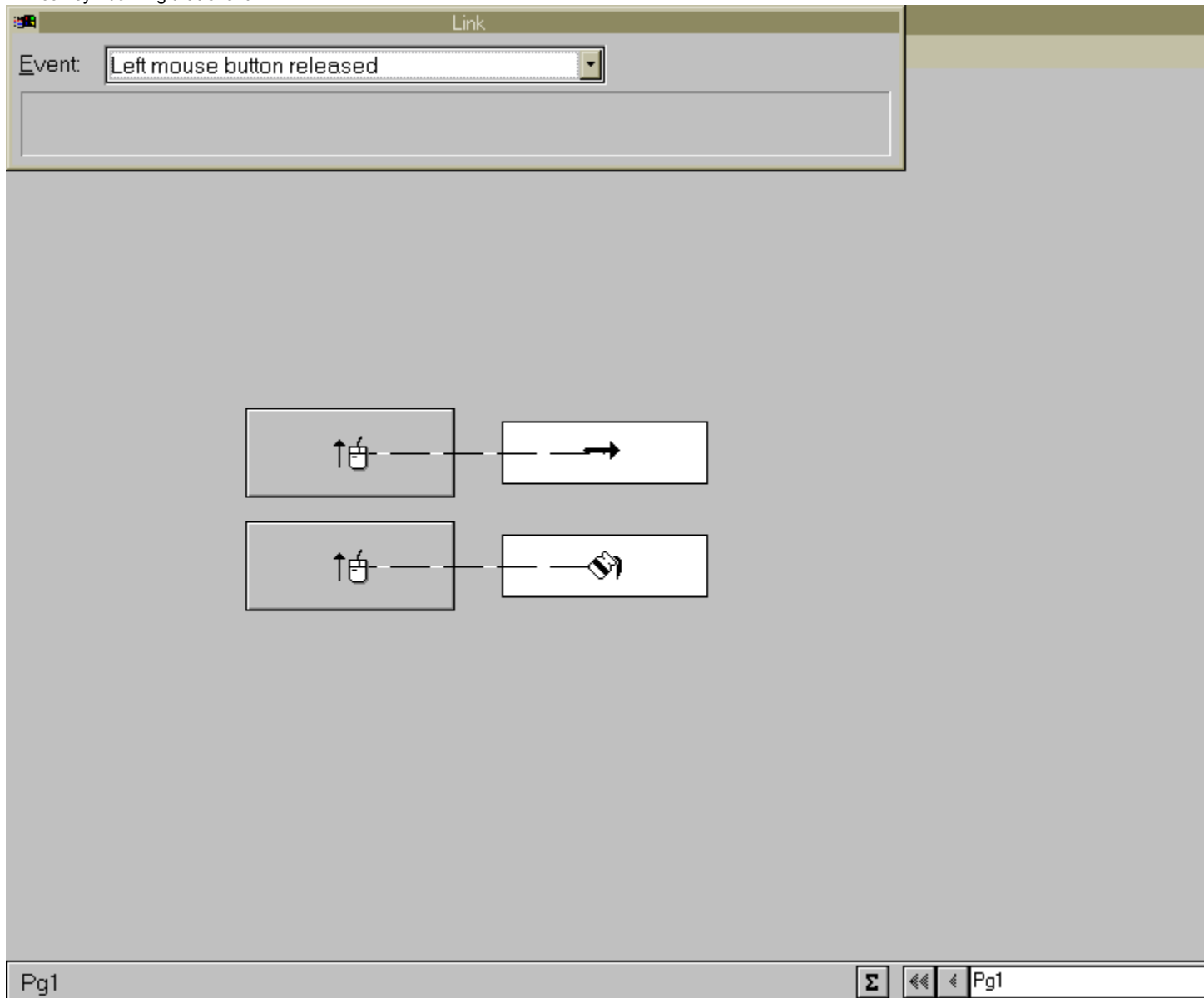


The cursor can take on other shapes, depending on your usage of MindMap and your selection in the [toolbox](#) or the **components** menu. When you click on **Objects**, a menu opens and shows a selected pointer. Whenever you have a different cursor and you need the pointer to select and edit components, click on that item. There are two other ways to change the cursor back to its initial setting:

1. use the shortcut key *CTRL+Y* or
2. click on the top button of the [toolbox](#) which shows the arrow.

## Link Option

This tool helps to control and evaluate different events that are connected with different components. Whenever you choose the Link option, a dialog box appears at the lower right side of the screen. Now you can select an event you want to control, e.g., **Left mouse button released** and all the components on the selected page which contain this event will show a special icon symbolizing that event.



If you select a component in this mode, then the list will display all links assigned to this component, followed by all links which point at the selected component.

The link component function offers a quick overview of selected events and allows you to search for events or to examine the program structure of an application.

If you want to see all events of a component, the link component function is also helpful. Select the event you want to analyze and click on one of the components. You will see that the link component dialog box lists all the events of the selected component. This function allows you to analyze and compare the events and messages of different components very fast and comfortably.

## Run Option

A click on the run mode icon in the toolbox switches MindMap to run mode and makes your application execute. The shortcut function key *F4* allows you to start or to exit the run mode again. The run mode will display the application as the user will see it. The edit mode will, on the other hand, display the application in its construction mode, allowing you to make changes to it.

The run mode helps to check your program without creating an \*.EXE-file. You can test certain steps of your development process by switching into the run mode before you save the application. If the result is as you expected it to be, you can save the application. If the result does not meet your expectations, correct your application or close the program without saving, so that you can start from the beginning.

## Zoom Option

Working exactly and precisely requires tools that help to position buttons in the right place or to create an output page with exact printing positions.



The zoom function is a tool that helps to look at things in different degrees of detail. The zoom factor depends on the detail you are choosing, i.e. the smaller the detail you wish to see, the higher the zoom factor must be.

Selecting the magnifying glass within the zoom mode changes the appearance of the cursor, i.e. the cursor becomes a magnifying glass. If you have selected the magnifying glass marked with a plus in the *zoom dialog box*, you can select a section to detail from a screen. Press the mouse button and pull the cursor over that part you want to zoom in on.

If you want to see the screen in the original size, you can click on the magnifying glass marked with a minus and the zoom factor regresses step by step. The fastest way to get the default resolution is to choose the 1:1 function.

