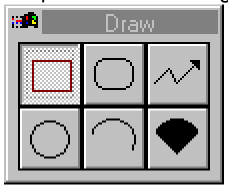
Component Types

This chapter provides a detailed description of each component and it attributes. Along with the component-specific attributes, all generic attributes relating to the component are described. Reference is also made to those attributes described in previous chapters. If a component registers its own events and/or messages, these are also described in this section.

Draw Component

You can select these components from the toolbox, as well as from the menu. After starting MindMap, you can see the following button on the toolbox, which represents graphical primitive components. If you select this component, you will be offered a choice among the six graphical primitives. After selecting one, its icon is used to represent the group until another one is selected.

Clicking on the Draw icon in the toolbox opens a window. This shows six drawing components, which are also called graphical primitives: a rectangle, a rounded rectangle, a line, a circle, an arc and a pie segment.

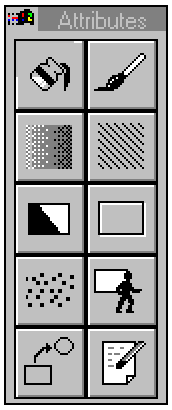


Graphical primitives are the simplest components in MindMap. In general, they are used for visual purposes. For example, if you wish to place a colored circle on the screen, you would select a circle and fill it with a color.

Rectangle

Selecting the rectangle within the draw mode changes the look of the cursor, i.e., the Pointer becomes a small cross with a little rectangle next to it. With the cross, you can position the rectangle on the screen. You can draw as many rectangles at one time, as you desire. If you want to leave the draw mode, just click on the Pointer in the toolbox. This feature of remaining in the selected mode until you change back to the pointer is called the sticky mode. This sticky feature is used in the draw mode and for command buttons.

Attributes



As you can see on the component attribute toolbox, this type of component does not offer any specific attributes.

Rectangle: Special Considerations

Rectangles are very common components in MindMap. Their usage is not restricted to graphical concerns. They are mainly used as:

- placeholders for subroutines,
- variables,
- and other development requirements.

Rectangles and the other graphical components can be sized by grabbing one of the drag marks while pressing the left mouse button. Move the mouse to the desired position on the screen and depress the mouse button. If you select a drag mark in the corner of a component, dragging the mouse will size its width and height. If you grab one of the other four drag marks, the components height or width will be sized. If you press the SHIFT key and perform the same action on one of the corner drag marks, the component will be sized proportionally.

Color

A useful feature is the transparent function. You can draw a rectangle, for example, that can be used as a frame for a bitmap. Select transparent as **Color** for the rectangle and you will see that the surrounding line of the rectangle can still be seen.

Lines

The second icon in the attribute box is the brush. This enables you to select a **Line** type and to define the color of a line. Select a color and change the line width from 1 to 5, for example. Be sure that the rectangle is a little bit bigger than the bitmap. Now you can position the rectangle over the bitmap so that it seems to have the frame of the transparent rectangle.

*Changing an attribute of the line is only valid if you have selected **No Frame** as the border style. This selection is the default setting.*

Rounded Rectangle

Selecting the rounded rectangle within the draw mode changes the appearance of the cursor, i.e., the Pointer becomes a small crosshair with a little rounded rectangle next to it. With the cross, you can position the rounded rectangle on the screen. You can draw as many Rounded Rectangles as you wish. If you want to leave the draw mode, just click on the Pointer in the toolbox. Again, this is a sticky-mode component and will remain active until you choose another component or the pointer.

Rounded Rectangle: Attributes

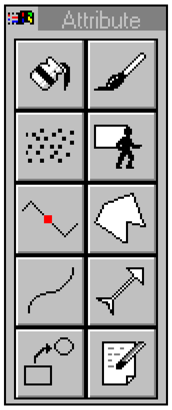
The attribute toolbox is almost identical to the one presented for the rectangle. The only difference lies in the fact that the rounded rectangle cannot have a border setting other than No Frame.

The available attributes of this component are described in the section
Format Menu.

Line

Selecting the line within the draw mode changes the appearance of the cursor, i.e., the Pointer becomes a small cross with a zigzag line next to it. With the cross you can position the line on the screen. A mouse-click starts drawing a line following your mouse movements. A second mouse-click sets an angle. A double-click with the mouse stops drawing a line. So if you want to draw a simple line, select line in the draw mode, position the cursor at the starting point of your line and click once. Move the cursor to the end point of your line and double-click. Once created, a line can be easily moved and resized by grabbing its handles. This is especially useful in wrapping a line around an component on the screen.

Line: Attributes



The attribute toolbox represented here, reflects the options for a line. It contains an attribute, Lines, represented by a paintbrush. By accessing this attribute, you can set the color, line width and line style. The button displaying a paint can icon permits setting the color fill of a   line, only if the line has been closed, turning it into a polygon.

Setting a Node

If you wish to define a new node on the line, begin by selecting the line itself. This is done by clicking on it once. Next, locate the position at which you wish to insert a new node. Now activate the attribute toolbox by clicking on the position for the new node. The attribute toolbox will appear. Now select the icon and a new node will appear on the selected line. You can select the node and move it around, thereby defining its position relative to the other nodes.

Deleting a Node

If you have drawn a polygon, for example, and you want to reduce the number of nodes, just click on one node, activate the Attribute toolbox and select the delete node icon. The node will disappear and a straight line between the next two remaining nodes will be drawn.

Fill

Sometimes you will need a graphical component that is not available on the toolbox. With the line component, you can draw the outline of the desired component. Select the line, activate the attribute toolbox and click on the line fill icon. As long as the line contains at least three nodes (which is equivalent to one angle), the two open ends will be connected, resulting in a convex and closed area which is filled with a white color. To change the color, just click on the paint can in the attribute box and select the appropriate color.

Note that lines that you wish to form a figure need not be connected at the ends. To create a triangle for, example, it is enough to draw two sides of the triangle with one line, and then use the line fill feature.

Fill Undo

The fill undo function removes the color filling of lines, so that you get back to the straight original lines, as you had drawn them.

Curved Lines

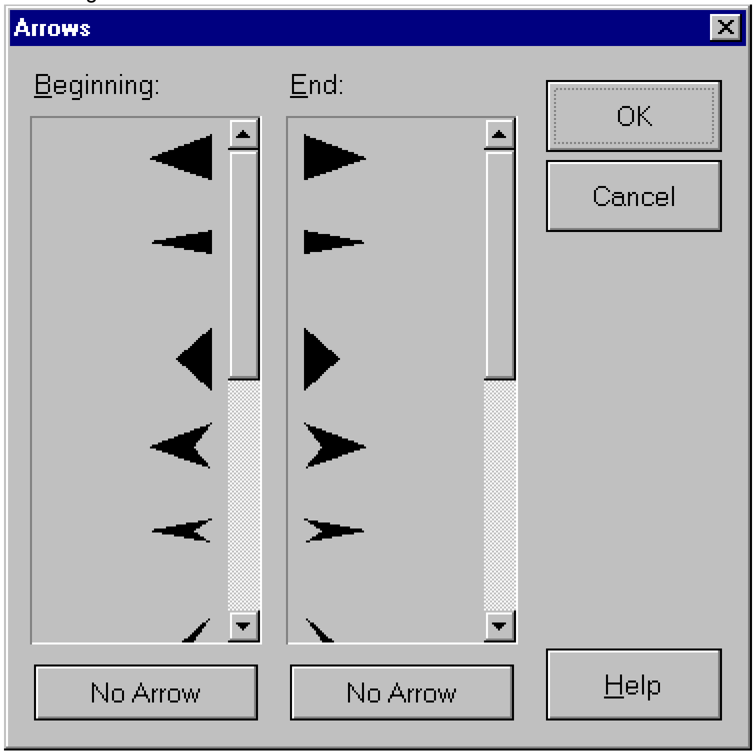
Normally, a straight line exists between two nodes, because the default for drawing lines is straight. MindMap allows you to change a straight into a curved line. Select the line and open the attribute toolbox. Click on the curved line icon. Each straight line segment will now have two crosses in the middle. If you select one of the crosses and then move it, the vector will be displayed as a curved line segment.

Straight Lines

This command is available for curved lines only and removes the curved lines created. If you have drawn a curved line and you want it straight again, you don’t have to draw a new line. Just select the line and click on the straight lines icon in the lines attribute box.

Arrows

A line can be changed to an arrow by defining special beginnings and endings for the line. Select a line and open the arrows dialog box from the attribute box.



Now you can select the look of your line by choosing the beginning signs and the ending signs. You can choose between ten different signs for the beginning and the end. A total of 10 tips and/or tails are available. If you wish to revert to the previous setting, click on the No Arrow-button.Consequently, the arrow icon is not available for filled polygons.

You can use a different color for the line and for the arrow tips. The color of the line is defined via the line attribute icon. The color for the arrow tip is set by using the color fill attribute.

Circle

Selecting the circle within the draw mode changes the appearance of the cursor, i.e., the Pointer becomes a small cross with a little circle next to it. With the cross, you can position the circle on the screen. You can draw as many circles as you wish. If you want to leave the draw mode, just click on the Pointer in the toolbox. This component operates in sticky-mode.

The draw mode allows you to create circles and ellipses with the same command. If you have drawn a circle, you only have to pull the selected sides of the circle to create an ellipse.

Attributes

The attribute toolbox for the Circle component is almost identical to the toolbox for the rectangle component. The only difference is again that it does not allow for the setting of a border attribute.

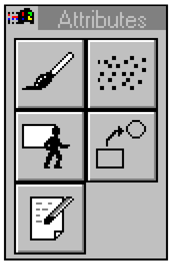
The available attributes of this component are described in the section
Format Menu.

Arc

Using the Pointer, you can position the arc and determine its size. With the mouse button pressed, you draw a rectangle where the arc is to be located. The size of the rectangle, and thus the arc, cannot be changed. The arc has a marker at the beginning and at the end, and these markers can be used to close or open the arc.

If you position the cursor on the marker of a selected arc, the cursor changes from a Pointer to a pointing hand. Press the mouse button and pull the arc to the desired size and shape, and release the button again. The rectangle specifies the size of the arc and the pointing hand draws the exact arc line, depending on your mouse movement. That means that you don't have to actually draw the arc. You only need to determine the beginning and the end, and MindMap draws the desired arc.

Attributes



Since this component is actually a line, the color attribute settings are disregarded. If you wish to define a line color for the arc, you should use the lines attribute

Please note the remarks on the other attributes in the sections above. They apply for the arc, too. The available attributes of this component are described in section Format Menu

Pie segment

With the Pointer you can position the pie segment and determine its size. With the mouse button pressed, you draw a rectangle where the pie segment is to be located. The size of the rectangle and, thus, the pie segment cannot be changed. The pie segment has a marker at the beginning and the end and these markers can be used to close or open the pie segment.

If you position the cursor near the marker of a selected pie segment, the cursor changes from a Pointer to a pointing hand. Press the mouse button and pull the pie segment to the desired size and shape, and release the button again. The rectangle specifies the size of the pie segment and the pointing hand draws the exact pie segment, depending on your mouse movement. That means that you don't have to draw the pie segment. You only need to determine the beginning and the end, and MindMap draws the desired pie segment.

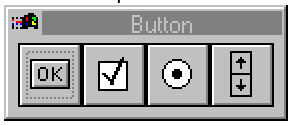
Attributes

The attribute toolbox for the pie segment component is identical to that of the arc. You can set the color attribute to fill the pie segment. The default fill color is set to white.
The available attributes of this component are described in the section
Format Menu.

Button Component

Buttons are very frequently used components and are, therefore, also placed on the toolbox. After starting MindMap, you will have this icon displayed on the toolbox. It too represents a selection of four components. If you click on this icon, another selection of icons is offered. Clicking on one of these icons will make it the representative icon on the toolbox, until you select a different button.

MindMap offers four different classes of buttons: command buttons, check boxes, radio buttons and scroll bars.

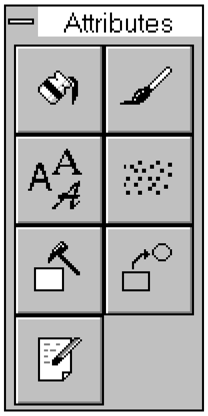


After making the selection, the cursor will appear as a crosshair with an attached button. Depending on which button was actually selected, one of the buttons will appear as part of the cursor. The cursor is modal (*sticky*) and will remain in the button draw mode, until you click on an the pointer or another icon button on the toolbox. This allows you to create as many buttons at one time, as you like.

Command Button

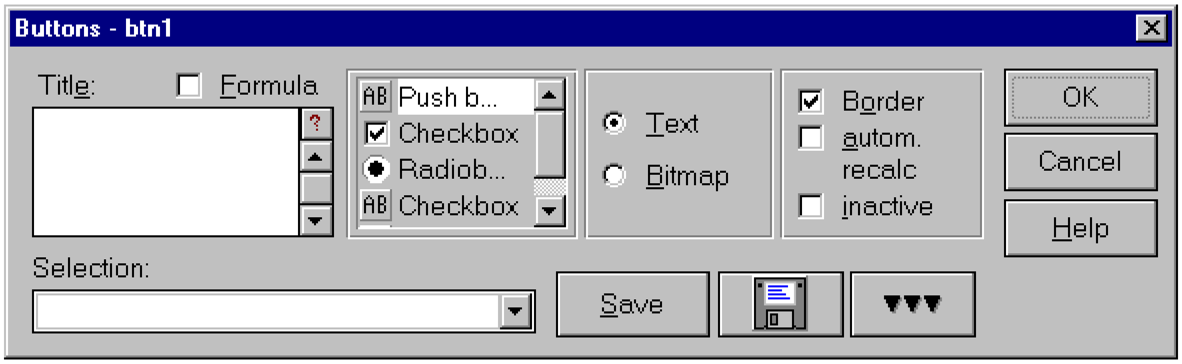
In MS-Windows applications, the command button is one of the most important components for navigating in the application and interacting with the user.

Command Button: Attributes



The attribute toolbox contains an icon offering a font selection, only if the command button has been set to have a text label. This is the default setting. If you change this to a setting where the command button contains a graphic, the font selection icon will not appear on the attribute toolbox.

You can set the color of the command button by using the color attribute. The line attribute permits the definition of the border width of the command button.



If you select **Text,** you can place text on the button. This is done in the **Title** field.

Note:
You can define accelerator keys for buttons. Accelerator keys are used in conjunction with the ALT key to select an action that is normally performed with the mouse. Pressing ALT and the desired character lets MindMap react as if the appropriate push button was pressed. To do this, include an ampersand ´&´ immediately before the character you wish the user to use as the accelerator key. Also, consider that on any one page, each such key should be unique.

If you set a check mark in the **Formula** field, the content of the title field will be parsed. That means that you can display the result of a calculation as the command button text or the content of input fields (which will be displayed if the title text contains the name of the input field), and the formula check mark is set. This is a very handy feature, if you want to dynamically define the label of a command button. This can also be used to build multi-lingual applications, for example.

Quite often, buttons don’t have a text label. In this case, you can include a graphic icon on the command button, which represents the intended function associated with the button. Select **Bitmap** in the button dialog box. The dialog box expands and shows an additional section where you can paint your own icon, by selecting a color for the left and right mouse buttons. Clicking with the left or right mouse button in the white icon display area paints a pixel in that color that you have assigned to the mouse button. By clicking on the **Delete** button, the icon display area will turn white and you can start painting an icon from scratch. It is usually much easier to click on the **Selection** list. MindMap offers a large list of bitmaps which you can show on a button, ranging from printers and arrows to question marks. The selected icon will be displayed in the icon display area.

Here, you can modify the icon to a customized button design. The **Save** button allows you to save your newly designed icon. Note that you must give a new name to a changed icon before you can save it. Clicking the button labeled with the **Diskette** icon opens the MS-Windows file dialog box. Here, you can select a bitmap from your system to be shown on the button. Note that the drawing area will display the bitmap converted to the color scheme shown in the left side of the icon display area. If you edit this bitmap, the whole bitmap will be converted. If you leave the bitmap unchanged higher resolution color bitmaps will also be displayed on the button.

The button labeled with the three **Triangles** opens or closes the icon display area.

The **Border** check box draws a border around the button. This is a default setting. If the option **Automatic recalculation** has been checked, then every time the button is pressed, all components on the same page will be instructed to perform a recalculation.

You can set a button **Inactive** if this is necessary in special situations. The button text or bitmap becomes grayed out and has no functionality.

If you wish, you can let a check box appear as a command button or as a radio button, and vice versa. This might be important in cases where you need to determine the state of a button. Standard command buttons do not retain their state. Check boxes and radio buttons do, but might not be the appropriate representation for what you want to display. In this case, simply redefine the appearance of a radio button or check box to be a command button.

Command Button: Additional Events

Event	**Explanation**
Button pressed	This event is relevant where you are using a push button as a checkbox or radio button. This is the default setting.
Button released	see above

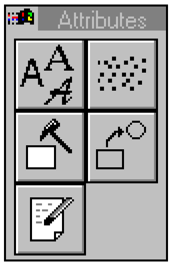
The common attributes are described in Format Menu

Checkbox

Checkboxes are a useful feature in the interaction with the user of an application. You can have a range of checkboxes in which you ask the user to make certain selections that influence the following steps of the program. Checkboxes differ from radio buttons in their multiple function. That means you can have more than one checkbox selected with a check mark, whereas radio buttons in a group only allow one selection. If you set an attribute for a group of radio buttons, then all members of the group receive the same attribute setting.

The common attributes are described in Format Menu

Checkbox: Attributes



See also: Command button: Attributes

The common attributes are further explained in Format Menu

Checkbox: Component Specific Attributes

The dialog box for the selection of the component specific attributes corresponds to the one for command buttons. The only difference is the selection of check boxes.

Check boxes can only display text, so the option to display a graphic is not available. You can type in text for the check box in the **Title** field. The text will be displayed in the right side of the box.

The other attributes are described in section Command button

Again, you can have a check box appear as a command button and vice versa.

Checkbox: Additional Events

see also Command button

Similar to the command button component, you can include an ampersand ´&´ within the text. The consequence will be that the letter immediately following the ampersand will be displayed with an underscore. Since text components do not allow user input at runtime, it does not make much sense to grant them focus. Accordingly, the focus is passed to the component next in the tab order. You can use this feature to position the cursor in an input field by assigning an ampersand to a text field which is used as the label for the input field. If you wish to have this feature for more than one combination of text/input fields, you must group each set separately and use unique speed keys for each group. If you only desire to have the focus set to the first input field, no grouping is required.

The dialog box for the selection of the component specific attributes is virtually identical to the one for command buttons. The only difference is the selection of check boxes as the type.

Radio button

Radio buttons are used for selections which are exclusive, meaning that making one selection will deselect any other selection. Exactly one selection can be made among radio buttons.

If you do not specify otherwise, all radio buttons on one screen are considered to be logically grouped. Thus, if you select any one of the radio buttons, any other radio button on the page is deselected. If you wish to have multiple groups of radio buttons on one page, then you must define multiple groups of radio buttons. This is accomplished by selecting the radio buttons you wish to have belong to one logical group, and then executing the menu option Properties | Group. Create different groups on one page by performing this same operation. Thus, you can select exactly one radio button per group.

Let's explain this by an example: You can paint two radio buttons asking for the gender of a person: male or female. If you want to ask for the age of that special person, you can suggest different ranges; for example, from 20 to 40, from 41 to 50, from 51 to 60 and from 61 to 70. That means you need four more radio buttons.

To answer these two questions requires six radio buttons. If you would use six Radio buttons on one MindMap page, MindMap would declare them as one group. That means that if you click on one Radio button and select it, any other button on that page that has been previously selected, would be deselected. The solution is to group the Radio buttons that ask for the gender in one group and to group the other four in another group. Now the user can click one button in the gender section and the selection of a button in the age section will not be affected.

The default setting of a radio button is 'not pushed'.
Its logical value is thus 'false'.

See also Properties | Group

Radio Button: Attributes

If you select the option Formula in the component specific toolbox, the text entry will be interpreted as a reference to a variable. This means that the text will not be taken literally, but rather as a variable containing the actual text which you want as a label.

See also Check box

The common attributes are further explained in Format Menu

Radio Button: Component Specific Attributes

See also Check box

Radio Button: Additional Events

See section Command button

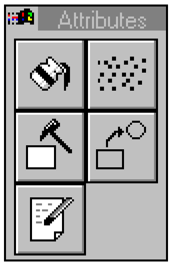
Scroll bar

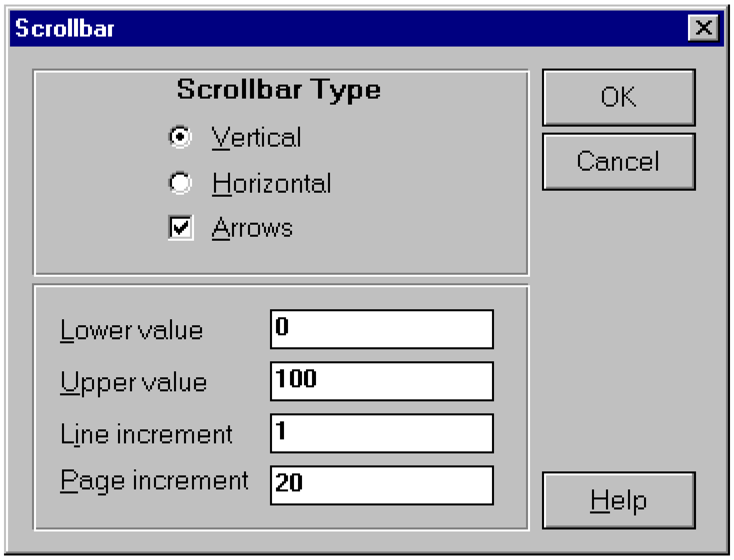
In the button dialog box, this icon symbolizes the scroll bar function. If you click on this icon, you can create scroll bars. Position the cursor at the starting point of the scroll bar and pull the rectangle, with the mouse button pressed, to the desired size. Depending on the way you draw it, the scroll bar will be either a vertical or a horizontal one.

If, when you place a scroll bar, you create a rectangle which is wider than it is high, a horizontal scroll bar will be created. Conversely, if you drag open a rectangle which is higher than it is wide, a vertical scroll bar will be placed on the screen. If you wish to change the orientation, you have two options.

The common attributes are further explained in Format Menu

Scroll bar: Attributes





Scrollbar Type: [✓ Vertical] Horizontal [✓ Arrows]

Lower value	0
Upper value	100
Line increment	1
Page increment	20

In the object specific attributes dialog box, you can change the **scroll bar Type** from vertical to horizontal and vice versa. Just click on the type of the scroll bar you want. You can further decide whether your scroll bar has **Arrows** at both ends or not.

You can specify an **Upper** and a **Lower value**. These are the internal start and end values of the scroll bar. The internal difference is always 100 percent. When you move the slider on the scroll bar, the internal value will change as a percentage of difference between the lower and the upper value. If you change the values, for example to years, let's say from 1901 to 1950, the scroll bar movement in run mode results in a recalculation of the internal value. If the scroll bar is moved by 2 percent, the internal value will change by 1 (*year*). You can assign this value to an input field where the movement of the scroll bar is shown as a percentage interpolation between the lower and the upper value. The value of the input field can then be used for further calculation or for displaying it to the user.

Line increment denotes the step size, when clicking on the scroll arrows on the scroll bar. Page increment specifies the step size when clicking on the scroll bar between the thumb and the scroll arrows.

If you want to use the message **Assign value** for assigning the internal value of a scroll bar to an input field, type the name of the scroll bar into the formula field of the assign value message in the **create and edit links** dialog box. Such a link could translate for example to "Event: Scroll change. Message: Assign value. Component: edt1. Formula: scrl1". Every modification of the scroll bar with the name "scrl1" would then display the internal value of the scroll bar in the input field "edt1".

The common attributes are further explained in Format Menu

The following list those events that are only available in the scroll bar links list.

Scrollbar: Additional Events

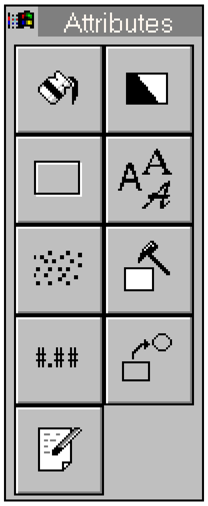
Event	Remarks
Row up/left	A message is only executed by mouse clicks on the upper (*vertical scroll bar*) or left (*horizontal scroll bar*) buttons with the arrows.
	The bar movement is dependent on the line increment.
	Note that clicking on the lower or rightarrow also moves the slider, but then no message is executed.
Page up/left	A message is only executed by mouse clicks on the upper (*vertical scroll bar*) or left (*horizontal scroll bar*) page. (*Page means the sector between the slider and the arrow buttons*).
	The bar movement is dependent on the page increment.
	Note that clicking on the lower or right page (*or on the arrow buttons*), also moves the slider, but no message is then performed.
Row down/right	A message is only executed by mouse clicks on the lower (*vertical scroll bar*) or right (*horizontal scroll bar*) page.
	The bar movement is dependent on the page increment.
	Note that clicking on the upper or left page (*or on the arrow buttons*) also moves the slider, but no message is then performed.
Move	Another way to move the bar is to move the slider with the pressed mouse button in the appropriate direction.
Scroll change	All the above mentioned features are assembled in this event. A message is performed whenever the slider of the scroll bar moves, no matter what caused the move.

Text Component

This component allows you to draw text fields on any MindMap page. This component can also be found on the toolbox as well as on the menu. It is represented on the toolbox by an icon displaying a capital T.

Selecting the "T" mode changes the appearance of the cursor; i.e., the pointer becomes a small cross with a "T" next to it. Using the cross, you can create and position a text field on the screen. Once you have placed and/or selected a text component, you will see a vertical line (*the insertion point*) inside the text component. It will be in the first column/first row. This line symbolizes the cursor inside the component and displays the current position.

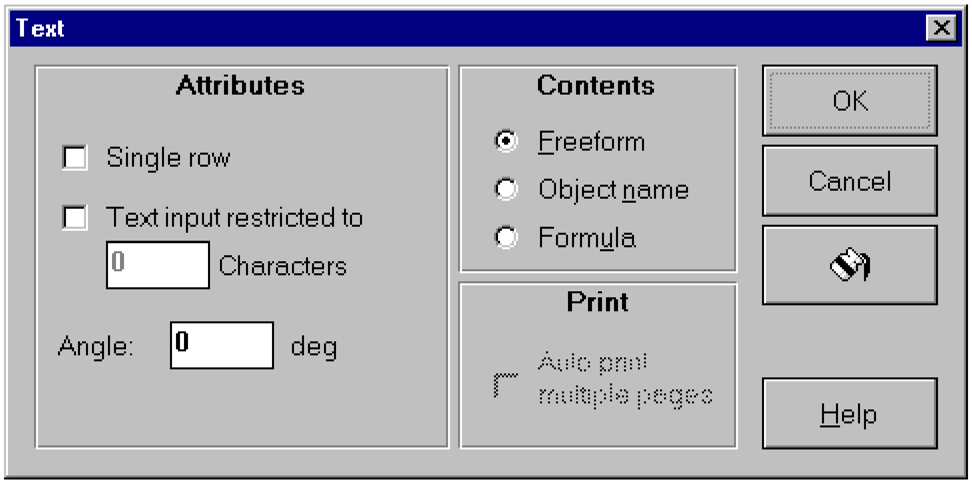
Text Component: Attributes



The common attributes are described in Format Menu

Similar to the command button component, you can include an ampersand ´&´ within the text. The consequence will be that the letter immediately following the ampersand will be displayed with an underscore. Since text components do not allow user input at runtime, it does not make much sense to grant them focus. Accordingly the focus is passed to the next component in the tab order. You can use this feature to position the cursor in an input field by assigning an ampersand to a text field which is used as the label for the input field.

Text Component: Component Specific Attributes



In the object specific attributes dialog box, you can determine whether your text display is restricted to a single row. You can also restrict the text input to a defined number of characters.

An additional feature for showing text on the screen is the **Angle** function. Here you can specify the number of degrees at which the text is rotated. You will need to experiment a little bit to understand the underlying logic of this feature. The angle has different effects, depending on the angle and the chosen text font.

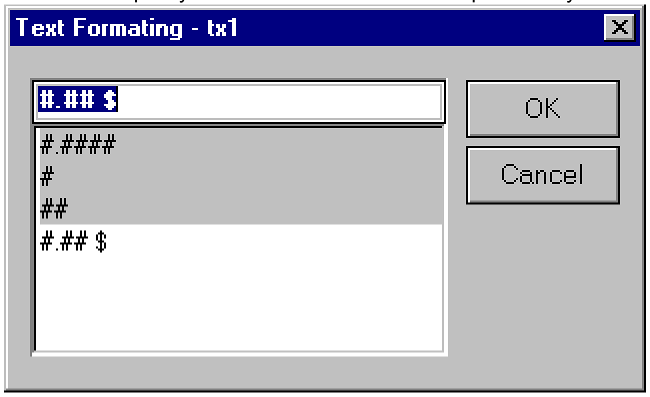
The text angle function only works with true type fonts. If you select a non true type font and want to angle the text, MindMap will ask you to select a true type font. Consider the fact that the ***Angle*** *influences the size of the text field. That means a horizontal text block may fit in the text field, but if you select an angle of 45 degrees, for example, the text that does not fit in the text field will be cut off. Make certain that the entire text block appears as you want it and adjust the size of the text field relative to the angle/font combination.*

The text fields appear as **Freeform** by default. If you change the content to **Object name,** MindMap will display the name of the component in the text field. If you select the option **Formula**, then you cannot enter text, as you are able to do when **Freeform** is selected. If you wish to have text appear in the text field, you must either use an Assign link, revert to the option selection Freeform, (see section *Links Messages: Assign Value,)* or enter the text into the parser window (*the button displaying a in the status bar*).

An additional option is available for text components placed on output page components. This permits a text field to wrap to a subsequent page, if its content - which is defined at runtime - is larger than the space provided by drawing the original text field on the output page component.

Text Component: Format Attribute

You can specify a format for the text field component. If you do so, the following dialog box appears.



This dialog box permits the definition of a specific format for numerical values. For example, you can determine where the decimal point is to be placed. You can also have a character, such as a dollar sign ´$´ included.

Text Component: Additional Events

No additional events are registered.

Text Component: Special Notes

When in edit mode, you can copy the content of the clipboard into a text field and vice versa. This is only possible if the text field has been set to freeform. Select the text field you wish to copy from and press *CTRL+Ins*. This copies the content of the text field into the clipboard. If you now wish to place the content into another text field, select it and press *SHIFT+Ins*, which is the shortcut key for inserting from the clipboard.

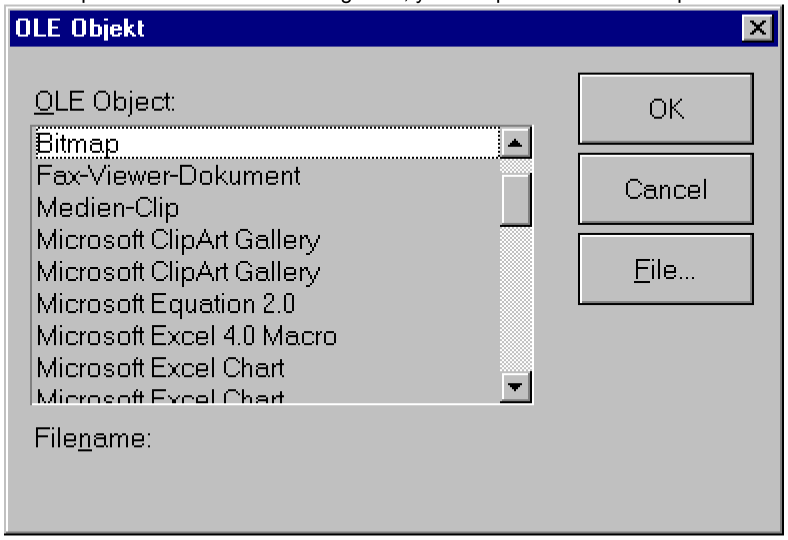
The CTRL+Ins and SHIFT+Ins keys are used to copy the selected contents of a text component into the clipboard. To copy the text component itself to the clipboard use the CTRL+C and CTRL+V keys. Note that in this case the contents in the clipboard are visible only to MindMap.

OLE Component

MindMap supports OLE 1.0 components

This feature cannot be found in the **toolbox**. It is only available in the menu **Objects | OLE…**

This feature allows you to place OLE components on any MindMap page. After selecting this function, a message box appears that asks you which kind of OLE component you want to interact with. Only those applications that have been properly registered by MS-Windows will be made available. Click on one of the choices offered. You can use the **File** button of the dialog box to search for the desired file or application on your hard disk. After this, or immediately after selecting the OLE component and without choosing a file, you can place the OLE component on the screen.


OLE Objekt

OLE Object:

Bitmap
Fax-Viewer-Dokument
Medien-Clip
Microsoft ClipArt Gallery
Microsoft ClipArt Gallery
Microsoft Equation 2.0
Microsoft Excel 4.0 Macro
Microsoft Excel Chart
Microsoft Excel Chart

Filename:

For additional information regarding common attributes, Format Menu.

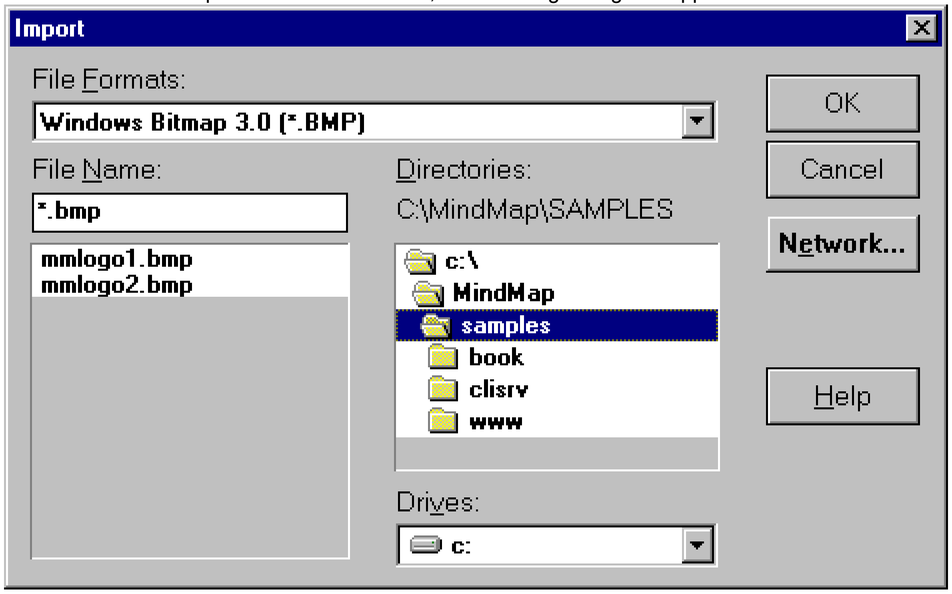
OLE Component: Object specific attributes

The object specific attributes of OLE components differ depending on the type of OLE component. A media player component, for example, can have "play" and "edit" functions, where as a text component may have totally different ones. You will have to explore this by yourself, using the documentation of the other applications.

The component specific attributes of an OLE component are identical to those listed in the message section of the links for an OLE component.

Graphic Import Component

The graphic component allows you to include various graphic files as part of your MindMap applications. Once you have selected this component from the toolbox, the following dialog box appears.



File Formats: Windows Bitmap 3.0 (*.BMP)

File Name: *.bmp

mmlogo1.bmp
mmlogo2.bmp

Directories: C:\MindMap\SAMPLES

c:\
MindMap
samples
book
clisrv
www

Drives: c:

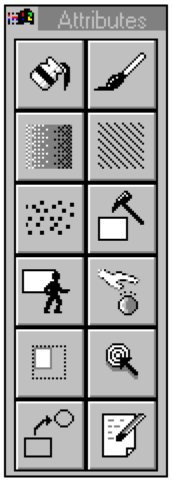
In this screen shot, the list containing the supported file formats has been expanded. The formats in this list, which are displayed without the asterisks in the front, are standard graphic file formats MindMap supports internally. Formats listed with an asterisk in the front are only available if you have installed other MS-Windows programs such as Microsoft™ Office, Lotus SmartSuite or a similar program belonging to this group, and have selected Import filters at the time of your original installation. MindMap recognizes these filters and will use them to read files that you may wish to import.

Only 16-bit versions of graphic import filters which follow the Lotus / Microsoft™ / Intel specification are recognized. When importing images, you may wish to use the standard integrated MindMap filters, since these support dithering.

Before you import a graphic, select the **File format** and then load the graphic. This may take time, depending on the size and the format of the graphic. Next, the cursor changes to a paint can and you can position the graphic on the page. There are two methods for placing the graphic. You can simply click once on the left mouse button. This will place the graphic in its original size and is the preferred method. The mouse position will define the top left corner of the graphic. You can also size the graphic when placing it, by pressing the left mouse button and dragging the mouse to represent the desired size. You can always change the size of the graphic later by either interactively sizing it, or by accessing its specific attributes and setting the desired size in terms of pixels.

If possible, you should avoid sizing a bitmap, since this might lead to undesirable visual effects due to the loss of pixels.
If a smaller representation of a graphic is requested in an application, we strongly recommend that you use one of the various available graphic tools to resample the image. Due to the fact that, by default, MindMap saves images into the application file, you can dramatically reduce the loading time of your application, if you reduce the size of the image.
This applies not only to the geometrical size of an image, but also to its color resolution. There is no need to import a 16-million color image, if it only uses 16 standard colors. Try to convert an image to a minimum amount of colors.
There are may shareware and/or freeware products available on the market to perform these tasks.

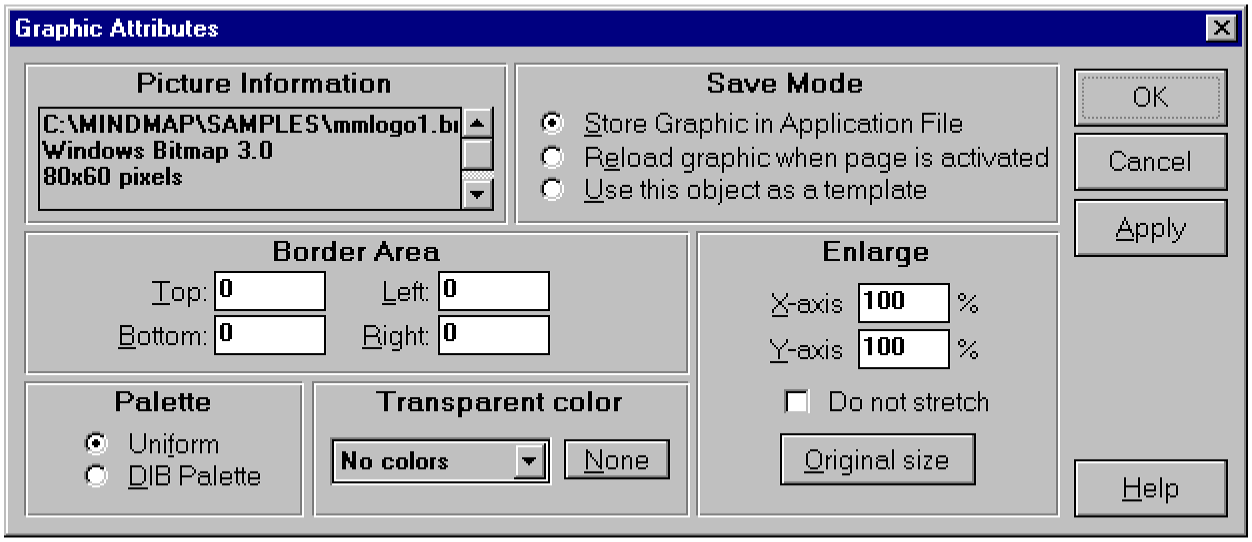
Graphic Import Component: Attributes



The attributes Color, Lines, Fountain fill and Fill pattern, all relate to the frame surrounding the graphic. The default width of this frame is zero. In order to set a frame, you must crop the graphic. This is accomplished by moving the mouse to a corner of the graphic. Press the *CTRL*-key and move the corner with the mouse. The cursor will change from the normal pointed index finger to now look like two right angles facing one another. Cropping retains the original size of the graphic and defines a new viewing mask. If you thus make the graphic larger than its actual size, it will receive a frame. If you make it smaller than its actual size, you will crop part of the graphic, while retaining its aspect ratio.

The common attributes are discussed in detail in Format Menu.

Graphic Import Component: Component Specific Attributes



Picture Information
C:\MINDMAP\SAMPLES\mmlogo1.bı
Windows Bitmap 3.0
80x60 pixels

Save Mode
[✓ Store Graphic in Application File]
Reload graphic when page is activated
Use this object as a template

Border Area
Top: 0 Left: 0
Bottom: 0 Right: 0

Enlarge
X-axis 100 %
Y-axis 100 %
☐ Do not stretch

Palette
[✓ Uniform]
DIB Palette

Transparent color
No colors

Graphic Information, in the object specific graphic attributes dialog box, gives you details about the graphic. You can see the name, the path, the filter and the size (*in pixels*) of the graphic. These parameters describe the imported graphic file as it will be saved into the application file. It does not reflect the actual settings of your graphic adapter.

If you import a true color graphic file on a computer that has a 256 color video adapter and move the application to a computer which is able to display 16 million colors the image in the application will automatically be displayed at its best resolution.

Save Mode offers three different ways to save the graphic. Which **Save mode** is the best, depends on your application.

- As the default, MindMap suggests **Store Graphic in Application File**. Each graphic stored in the application file will enlarge this file. Keep in mind that the size of graphic files is generally large and this will tend to dramatically increase the size of your application.Also note that MindMap saves all types of pixel-oriented input formats as if they were bitmap files. Input formats using extreme compression algorithms (TIFF, JPEG) may increase the size of the application file dramatically.
- **Reload graphic when page is activated** is a good possibility, if you can be sure that the graphic will actually be available at the specified location when the application is eventually executed. The benefit of this is that the application file does not contain the graphic file(s) and is thus generally much smaller. Deployment of your final application will require careful thought, if you choose to employ this method, but it has several advantages.
- If you have a page in your application where different graphics are to be shown (at one and the same screen position), you can use the function **Use this object as a template**. By using drag & drop between an input field and a graphic template, you can assign a new graphic to this template. This allows you to have a single graphic component for graphics display in your application and, via the drag &drop function and the template option, many graphics can be shown. For example, imagine a list of graphics. Whenever the user selects a file name, the graphic is displayed on the screen.

Border Area refers to the frame created by cropping a graphic. You can also create it by making the appropriate settings.

Palette (applicable only on 256 color video adapters): Here you can select whether a graphic uses its predefined **DIB** (*Device Independent Bitmap*) palette or a MindMap-assigned **uniform** palette to show the graphic. If you want to show more than one graphic in 256 color mode on one page, you have to choose the uniform palette and the graphics will be shown with a common palette. If you choose the DIB palette, the graphics will switch between different palettes and the palette of a graphic which has the focus will be applied to other graphics on that page. Sometimes it may be better to use the DIB palette, as the graphics will have a better display quality. Due to the wide variety of possible graphics combinations, you must experiment with your individual situation for the best choice of palette.

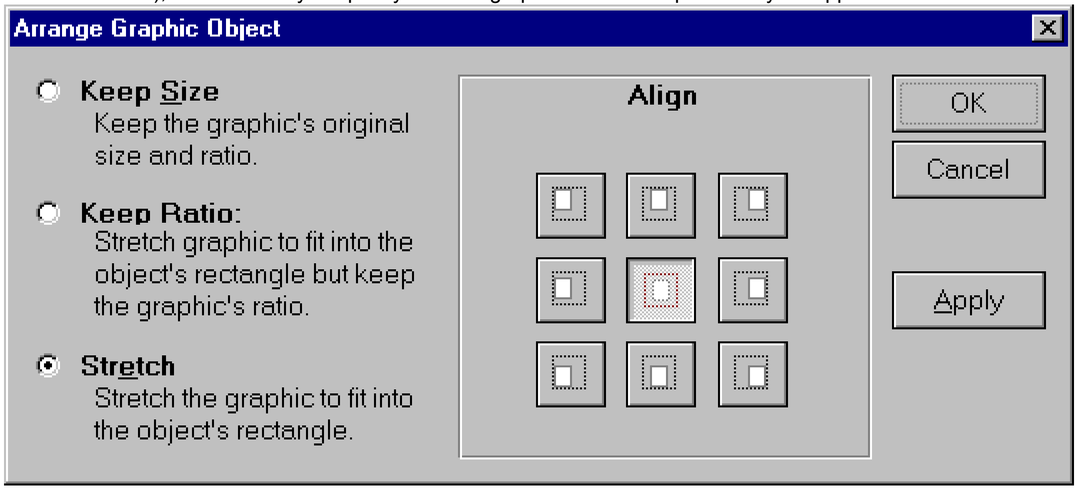
MindMap uses a technique called *dithering,* to map the color pixels of an image to a common set of colors (*a palette*). The benefit is that multiple images can be displayed simultaneously on the same page. The disadvantage is that this technique increases the load time of an image.

Transparent color gives special effects to graphics. You can select transparent as the color, so that the background of the graphic will show through. Note that this is not applicable for images which consist of more than 256 colors.

You can **Enlarge** the graphic, as measured in percent. MindMap resizes the graphic so that it will become bigger or smaller. You can set that the graphic will not be stretched when you change the size. This means that its height/width ratio will not change if you type in another size in either the X-axis or the Y-axis fields. Clicking on the button **Original size** sets the X-axis and Y-axis values back to 100 percent and the graphic will be shown in its original size. Please note that these settings depend on the **Arrange Graphic Object** settings.

Arrange Graphic Object

Especially when you work with templates (Object specific Attribute
Save mode), it is necessary to specify how the graphics should be placed in your application.



Arrange Graphic Object

○ **Keep Size**
Keep the graphic's original size and ratio.

○ **Keep Ratio:**
Stretch graphic to fit into the object's rectangle but keep the graphic's ratio.

● **Stretch**
Stretch the graphic to fit into the object's rectangle.

Keep Size: This function keeps the graphic’s original size and ratio. Whenever you have different sized graphics, whose size has to be fixed, you should set this radio button to **on**. Your graphic will not be changed and will be displayed in its original size.

Example: If your graphic is larger in size than the template and you have clicked **keep size,** the template will be overlapped by the graphic.

Keep Ratio: Whenever you drag & drop a graphic to a template and the size of these components vary, you should click on **keep size** or **keep ratio,** depending on your graphic. The function **keep ratio** stretches the graphic to fit into the component’s rectangle, but keeps its ratio.

Example: If your graphic is larger in size than the template and you have clicked **keep ratio,** MindMap reduces the graphic to fit into the template, i.e., the graphic becomes smaller in size, but keeps its ratio.

Stretch: Sometimes it is very useful to change the size of a graphic, so that you can use one graphic on different pages in different sizes. You can also use this feature if you intend to display various graphics in the same screen position and wish to have them all appear in the same frame. In that case, one can use one big graphic and reduce it to a smaller one on another page. (*This results in a better quality than blowing a small graphic up to a big one*). This function **stretch**es the graphic so that it fits into the component’s rectangle.

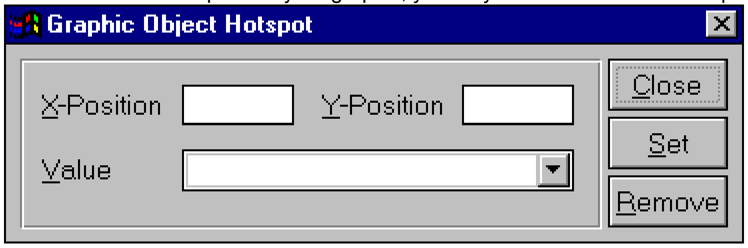
Example: If your graphic is larger in size than the template and you have clicked **stretch,** MindMap resizes the graphic to fit into the template, i.e., the graphic becomes smaller in size.

Align: Whenever the size of the template differs from the size of the graphic which is to be displayed, you should set the alignment. You can choose between nine different positions from the upper left corner of the rectangle to the lower right position.

The function **Apply** is used for testing the Alignment. This way, you can check which function best fits your needs.

Graphic Component Hot spot

MindMap offers the possibility to put messages on top of graphics. This is very useful for interaction with the user. If you want to access certain parts of your graphic, you may be able to use the hot spot function.



After you have opened the graphic component hot spot dialog box, you can click with the left mouse button on a part of your graphic. Clicking on the graphic while the hot spot dialog box is open, changes the color of the hot spot and displays its position and value in the window. In the dialog box you can see the **X-** and **Y-position** of the hot spot. Select all the desired hot spots you want and click on **Set** each time. By clicking on Set, the corresponding coordinates are entered in the list. If you wish to assign a character string to a coordinate (*such as a city name on a map*), enter the string in the Value field and then click on the Set button.

In the **Value** list box, you can see the coordinates or the character string assigned to the hot spot.

Example: If you have imported a graphic of a map with a number of cities on the MindMap page and you want to produce special information about these cities in Run mode, click on one of the cities to create a hot spot in Edit mode. Now you can name the hot spot via the value in the list box. This value can be assigned to an input field. Whenever the user clicks on one of the hot spots in Run mode, the given name of this hot spot is shown in the input field. You can then connect this input field with a link, for example, to a list box or another component that can display additional information corresponding to the selected hot spot.

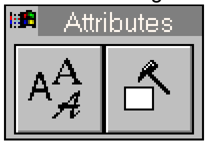
When the user clicks on the hot spot, the component itself will receive the value associated with the hot spot. The selected hot spot will display in green; all others in red. If you select one hot spot, all others are deselected.

Graphic Import Component: Additional Events

No additional events are registered.

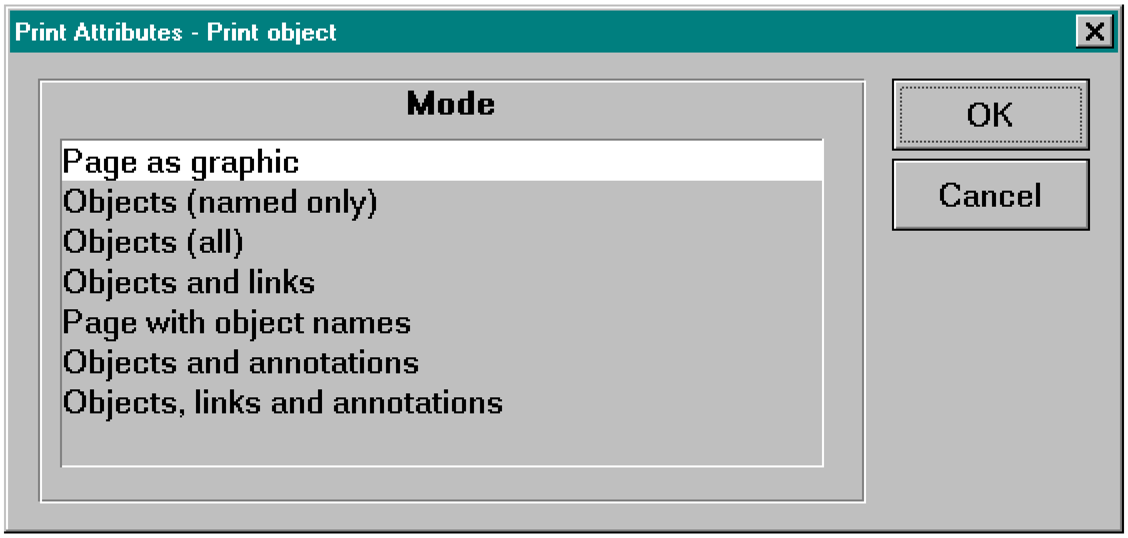
Printer Component

This component is only accessible through the menu and is grayed out in "normal" MindMap applications. It is only used in **Printer Layout files.** The Printer component is used for printer templates only. The Printer Component specifies the layout of the Printer Template. This template is stored in the file, STANDARD.MMP. The Printer component is a helpful tool for documenting and analyzing your application.



For further information concerning common attributes see Format Menu

Printer Component:Component Specific Attributes



MindMap offers the following modes for this component:

Mode	Description
Page as graphic	Prints the page as it is displayed
Objects (named only)	Prints a list of all components with their names
Objects (all)	Prints all components with name and symbol
Objects and links	Prints the components and links
Page with object names	Prints the names of the components of one page
Objects and annotations	Prints the components with name, symbol and annotations
Objects, links and annotations	Prints the page, the links, the components and annotations.

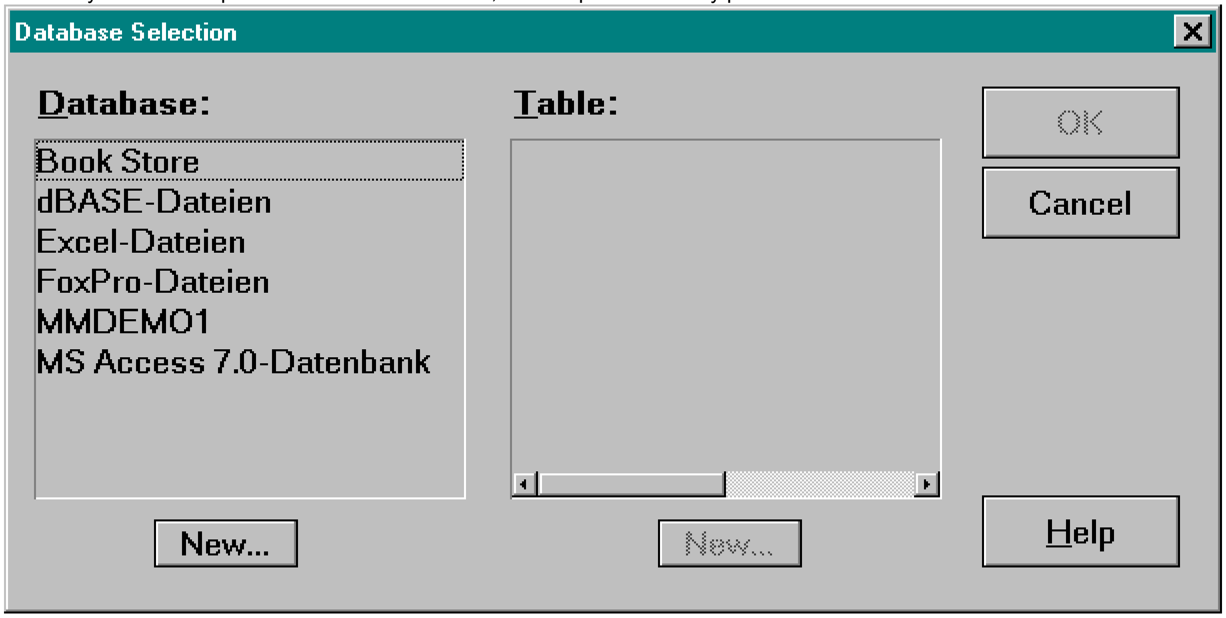
Database Manager Component

A powerful feature of MindMap is the database component. You can connect a database with the Windows ODBC (*Open Data Base Connection*) driver. This component permits you to access existing data sources and, in some cases, to create new sources.

Please be sure that the necessary database drivers are installed and correctly configured. These are either already installed or you opted to install them during the MindMap setup routine. If not, you can install the drivers using the ODBC Administrator tool, which should be located in your \WINDOWS directory.

If you click on the database icon in the toolbox or the menu option **Objects | Database Manager,** a dialog box opens where you can select the database driver, normally **ODBC**. MindMap offers you an option to install additional modules. These modules allow you to access other systems and data sources, which might not be available via the ODBC interface. If you have installed supplemental MindMap drivers, data sources accessible through these drivers will also be listed.

If only one MindMap database driver is installed, MindMap automatically proceeds to the **Database Selection**:



In this dialog box, you can select a **Database** and the corresponding **Table**. The tables will be listed as soon as you select a database. Please note that, while your application may require multiple tables to operate as you intend, MindMap allows you to place only one table at a time. You may have many tables placed and open at any time.

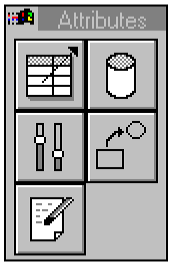
Please note that the names which appear in the ODBC data source list are not necessarily identical to the actual database names. The ODBC Adminstrator program allows you to assign names of your choice to database files.

The **New** buttons allow the setup of new databases and tables. In the first dialog box, you select the special ODBC driver for the database, e.g., Access Data (*.MDB) or dBase files (*.DBF). The following dialog box helps to choose the appropriate database or to create a new one. These dialog boxes are described in more detail in the documentation dealing with the ODBC Administrator. The ODBC Administrator can be accessed through the MS-Windows Control Panel or through the MindMap program manager group. Please note that not all database drivers permit you to create new databases. In addition, depending on the database and your company policy, you may not have creation permissions. Please consult your database administrator for more details.

After you have selected the database and the desired table, the cursor changes to a cross with the database icon next to it.

You can position the database component anywhere on the   page. The indented component display shows the driver, the name of the database or data source, and the table name. The Database wizard dialog box then appears. You may want to use the Wizard to automatically set up connected input fields or a data table, as well as some buttons. Refer to section **Database wizard** for more information. We assume here that you just click "OK".

Database Manager Component: Attributes

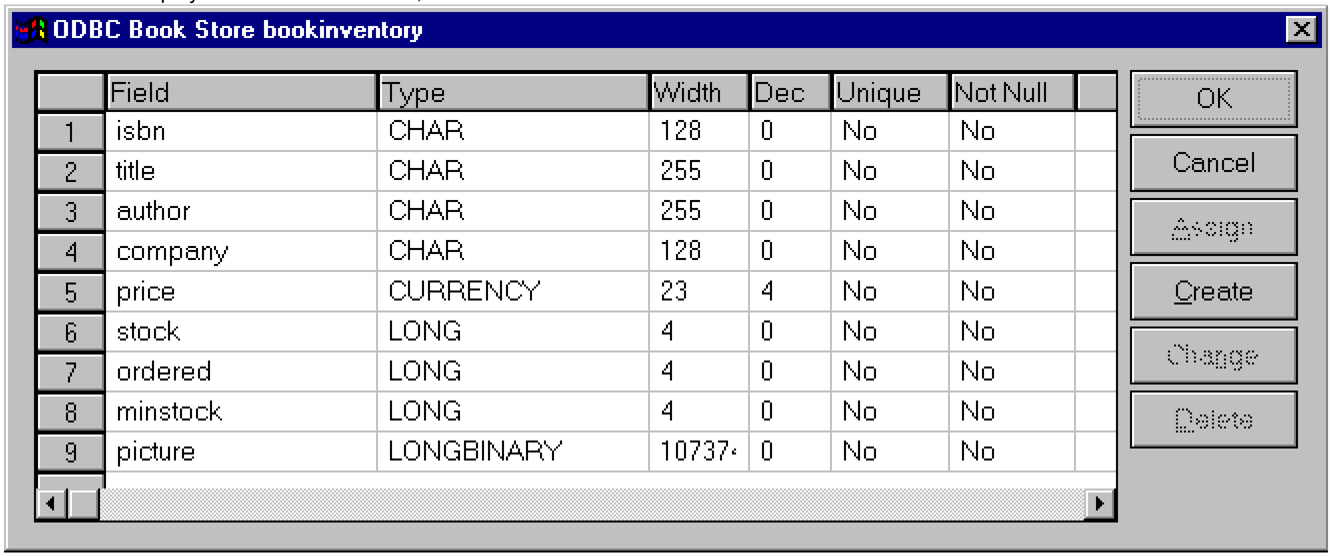


The common attributes are described in Format Menu

Note:
This component is only visible in edit mode. One you switch into run mode this component becomes invisible to the user.

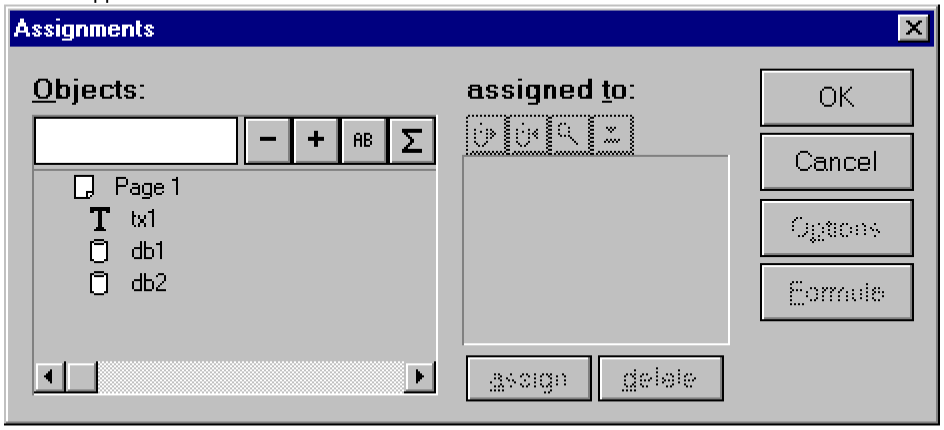
Table Structure

A click on this attribute of the database component opens a dialog box representing the structure of the database table. All fields are displayed with their definitions, as defined in the database itself.



	Field	Type	Width	Dec	Unique	Not Null
1	isbn	CHAR	128	0	No	No
2	title	CHAR	255	0	No	No
3	author	CHAR	255	0	No	No
4	company	CHAR	128	0	No	No
5	price	CURRENCY	23	4	No	No
6	stock	LONG	4	0	No	No
7	ordered	LONG	4	0	No	No
8	minstock	LONG	4	0	No	No
9	picture	LONGBINARY	10737	0	No	No

You can assign these fields to MindMap components such as input fields or data tables, for example. Click on a row number (*first column*) in the table (*representing a field*) and the row is now highlighted. Select **Assign,** so that the **Assignments** dialog box appears.

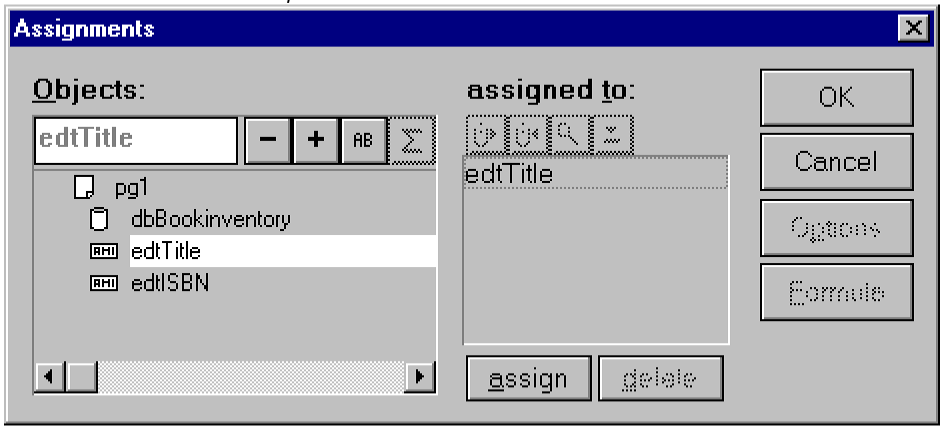


The simplest and, therefore, most often used technique for assigning components to database fields is the so called rubber band method. Open the table attribute of the desired database component that has already been placed on the page. Click on the row number of a database field (the row will thus be highlighted) and keep the left mouse button pressed. Now, drag the cursor to the component that will be assigned to the field. As soon as the cursor leaves the dialog box, you see the rubber band at the cursor. Position the cursor over the component and release the mouse button. If you successfully established the connection, the associated component name can be viewed in the last column of the assignment dialog box. All assigned components are listed there. Normally, you must scroll over to the right to view this column.

In the left side of the **Assignment** dialog box is a list with the components of your application. Select the component by clicking on it and press the assign button or just double click the component. Now you can see the component name in the

assigned to list.

You can assign multiple components to one field. All of them will then be listed. Each component can have individually set options,   but only one component can have write access. MindMap helps to prevent conflicting write access by prompting the user to choose which component shall have write access.

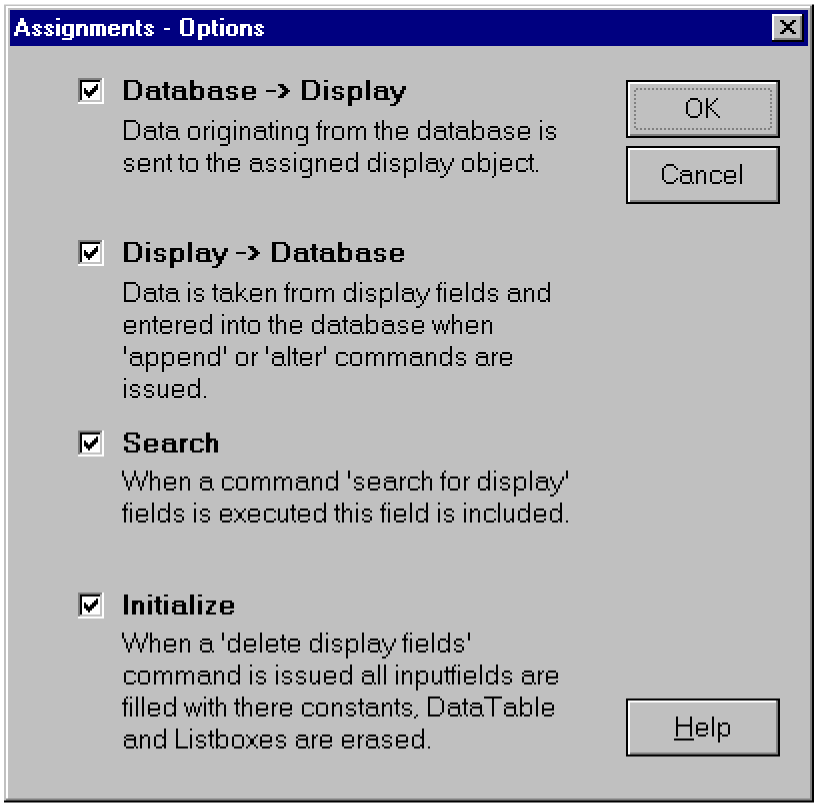


Objects: edtTitle

assigned to: edtTitle

If you select the component in the **assigned to** list of the **Assignment** dialog box, four small buttons above the list will be activated. A click on one of them will change the assignment option. You can reach them all together via the **Options** button. The following table shows the assignment options. The buttons represent the settings in the following dialog box, which were called via the Options button.

Function	**Icon**	**Description**
Database Display	▶	Data originating from the database is sent to the assigned display component (read access).
Display Database	◀	Data is taken from display fields and entered into the database when *insert* or *update* commands are issued (*write access*).
Search	🔍	When the command *search for display fields* in the message part of the link ‘Database’ is executed, the database searches for the contents of this field. This means that the contents of the WHERE clause in the SQL statement will be included. Please note that for the input field component, you may select from a list of predefined search methods.
Search	🔍	When the command "search for display fields" in the message part of the link ‘Database’ is executed, this field is included. This means that the contents of the WHERE clause in the SQL statement will be included.
Initialize	⏏	When a *Clear display fields* command is issued, all input fields are filled with their given preset values. The contents of data tables and list boxes are erased.

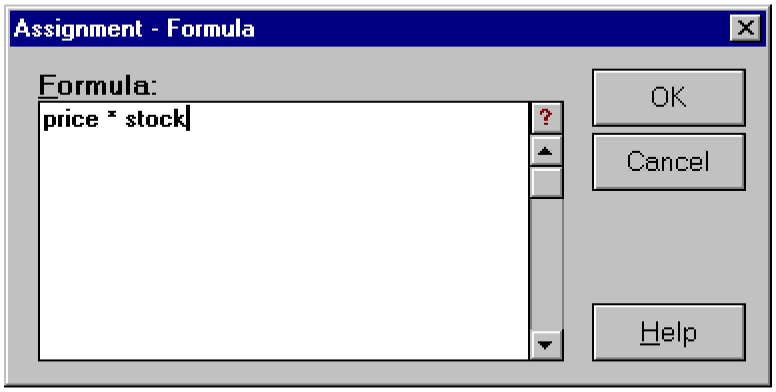

Assignments - Options

☑ **Database -> Display**
Data originating from the database is sent to the assigned display object.

☑ **Display -> Database**
Data is taken from display fields and entered into the database when 'append' or 'alter' commands are issued.

☑ **Search**
When a command 'search for display' fields is executed this field is included.

☑ **Initialize**
When a 'delete display fields' command is issued all inputfields are filled with there constants, DataTable and Listboxes are erased.


Assignment - Formula

Formula:
```
price * stock
```

The **Formula** button in the **Assignments** dialog box contains the name of the database field, by default. Here, you can type in a formula that will be parsed.

The database component will then evaluate the given expression for each record retrieved from the database and assign the result to the corresponding MindMap component.

You may want to use this function, if the database source cannot perform a similar task. To increase the overall performance of database access, you should attempt to let the database perform calculations like this. This is especially true for client/server SQL database systems.

Please note that you cannot update into the associated table field, regardless of whether you let MindMap do the calculation or the database system.

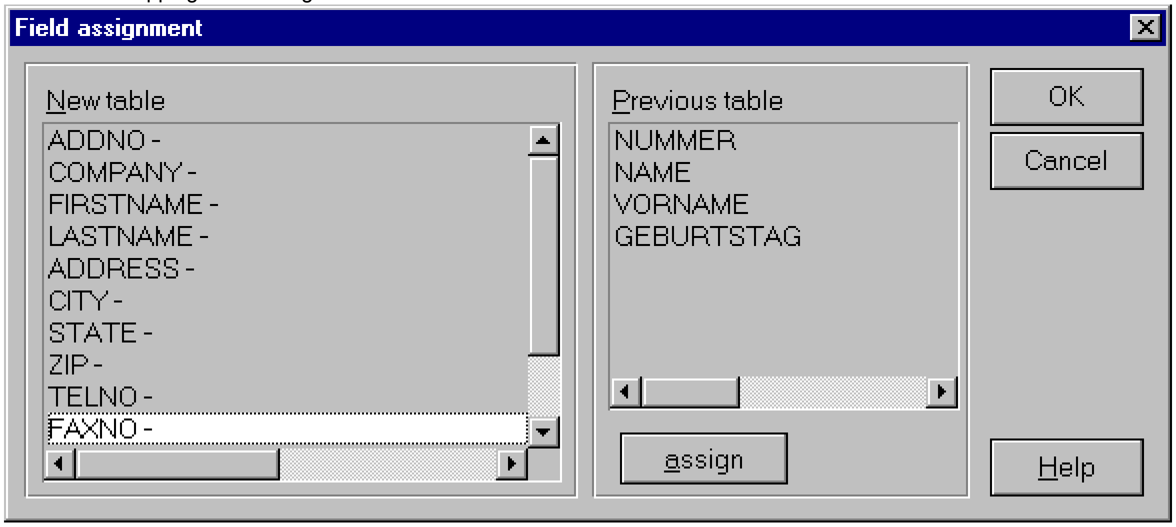
If you want to **Delete** the connection from the database to one of the components in the list, select its name and the **Delete** button will be activated so that you can delete the assignment.

You may also create new fields, as well. Click on the appropriate button and a new row will be added at the end of the structure table. Clicking in the appropriate columns allows you to enter the name of a new field, along with its definitions (*width, type, etc*.).

Database

You can use this button to replace one database with another one. This may become necessary if the structure of the database tables has changed. This works as described before with a new database, except that a field assignment dialog box appears where you can assign the fields of the **New table** to the **Previous Table**. When you load the new table, MindMap automatically suggests new assignments. This is only possible if both tables (*the one being replaced and the replacement*) have corresponding field names. You can also override the definitions or make the necessary assignments manually, should the field names be different.

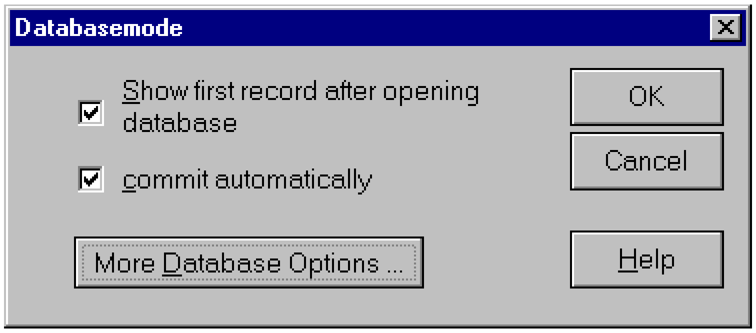
Example: This function is quite useful in the case where you are developing an application on your PC and running against a PC-based database. You then want to move it to the production system, perhaps an Oracle or Informix database, and deploy it in the organization. This is where you would re-map the database tables to reflect the new database and its tables. If you have the same field names in the tables of both databases, they will map across automatically. If not, you will have to do some manual mapping of the assignments.



New table	Previous table
ADDNO -	NUMMER
COMPANY -	NAME
FIRSTNAME -	VORNAME
LASTNAME -	GEBURTSTAG
ADDRESS -	
CITY -	
STATE -	
ZIP -	
TELNO -	
FAXNO -	

Database mode

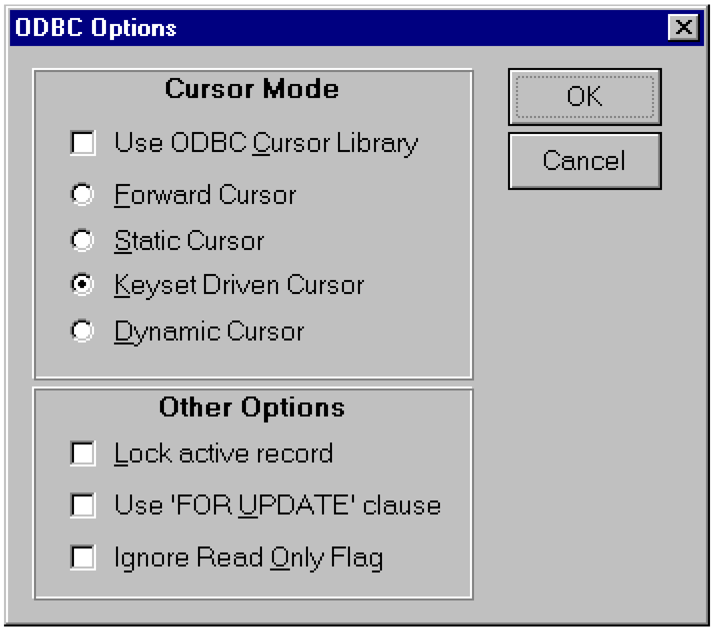
An additional attribute of the database component is the Database mode.


Databasemode
Show first record after opening database
commit automatically
More Database Options ...
OK
Cancel
Help

- ***Show first record after opening database****:*
  If this option is checked, MindMap will automatically select the first record that the database system has found in the appropriate table and display its contents in the associated input fields. If a data table has been connected to the database component, all records in the table will be displayed. Obviously, if the table contains a huge amount of records, you may want to uncheck this option. Depending on the kind of database system you are using, it may take a very long time to create a result set containing all records in the table. Better uncheck this option and create a result set of a reasonable size by issuing the database message Search for Fields or directly execute a SQL statement if supported by the database.
- ***Commit automatically:***
  This option is supported only in the case of transaction oriented SQL database systems. These databases allow you to "undo" the execution of certain database commands (such as insert, update or delete). This is called a ***rollback*** *in the terminology of database systems. Writing multiple transactions persistently to the database file is called a* ***commit*** *operation. The* ***commit automatically*** *option specifies that changes to a database table are automatically commited. Consequently, rollback operations are not applicable in this case.*

Note that ODBC database drivers like dBase, Text or Excel do not support commit/rollback commands. Only Microsoft™ Access and most of the available SQL databases support this feature.

You can set **More Database Options** with the appropriate button.


ODBC Options

Cursor Mode

- [ ] Use ODBC Cursor Library
- ( ) Forward Cursor
- ( ) Static Cursor
- (●) Keyset Driven Cursor
- ( ) Dynamic Cursor

Other Options

- [ ] Lock active record
- [ ] Use 'FOR UPDATE' clause
- [ ] Ignore Read Only Flag

Database Manager Component: Additional Events

In addition to a number of standard events, the database component has the following additional events:

Event	Description
Record loaded	As soon as a record is loaded, this event is triggered.
Refresh data fields	A refresh of data fields triggers this event.
Delete failed	This is a very useful event that shows if there are problems with the deletion of records.
Insert failed	This event is if an insertion into the database is unsuccessful .
Update failed	This event is triggered if an update of the database is unsuccessful .

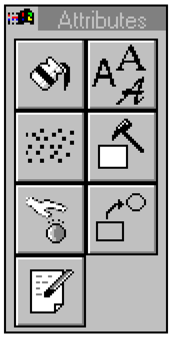
Database Manager Component:Special Considerations

Please refer to the special parser functions associated with the database component. (Parser)

List-/ Combo box Component

List boxes are one of the most often used components in MS-Windows. They are essential for selecting or displaying entries from a list. MindMap offers several types of list boxes that each differ in appearance and functionality. The desired type can be determined in the component specific attributes.

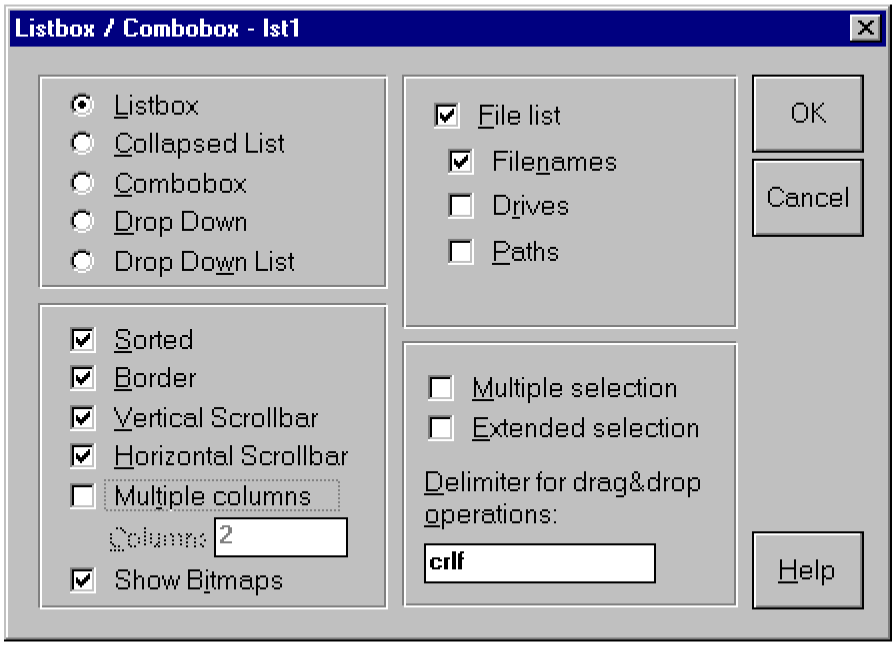
List-/ Combo box Component: Attributes



The Color attribute determines the background color of the list box. The Font attribute permits you to select the font and the color of the font.

The other common attributes are described in: Format Menu

List-/ Combo box Component: Component Specific Attribute



Listbox / Combobox - lst1

- [✓] Listbox
- [ ] Collapsed List
- [ ] Combobox
- [ ] Drop Down
- [ ] Drop Down List

- [✓] Sorted
- [✓] Border
- [✓] Vertical Scrollbar
- [✓] Horizontal Scrollbar
- [ ] Multiple columns
- Columns: 2
- [✓] Show Bitmaps

- [✓] File list
  - [✓] Filenames
  - [ ] Drives
  - [ ] Paths

- [ ] Multiple selection
- [ ] Extended selection

Delimiter for drag&drop operations: crlf

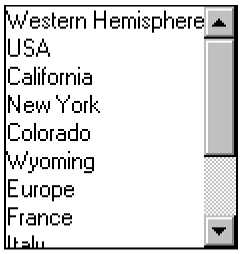
Classes of List boxes

There are a total of 5 different types of list-/combo boxes. These are:

- *List box*
- *Collapsed List box*
- *Combo box*
- *Drop down*
- *Drop down List*

In general, these various types of the same component differ in how they display data and how they accept input from the user. A list box displays data at a single level and does not accept any input from the keyboard. You can drag&drop data to and from it, though. A collapsed list box is the same, with one exception - it can display data at multiple levels. If the data has a hierarchical structure to it, similar to folders and files in a file system, this type of list box can reflect this structure. It does so by prefixing a row with a ´+´ to symbolize a level containing lower levels. A combo box can be considered as a list box with an input field tagged to the top of it. This permits keyboard entry. A drop down only shows the current selected entry of the component. All other entries are hidden behind an arrow icon to the immediate right of the display field. Once the arrow icon is pressed, the list literally drops down to show additional entries. Finally, the drop down list differs from a drop down, in that it will not permit keyboard entry.

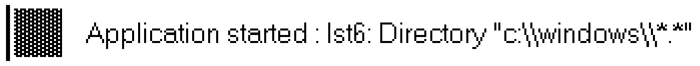
List box



The figure displays the standard list box. Its default setting specifies a visible border. If more entries exist than the list box can display with the selected **Font**, a vertical scroll bar will be automatically generated. It will only be generated if the check box **Vertical Scroll Bar** has been selected in the component specific attributes. MindMap automatically defaults to this option. If you desire, you can also have a horizontal scroll bar generated.

If you wish to have the entries displayed in sorted order, then you must make the appropriate selection in the component specific attribute. The list box also offers the option of displaying values in multiple columns. Again, the number of columns is set in the component specific attribute dialog box.

You can also specify that the list box is to be used to display files. The component specific attribute settings permit you to select whether you wish to have only file names, drives, or paths displayed. If you select this attribute, MindMap offers a dialog in which you can specify which files you would like to have displayed. The result of this dialog is a link:


```
Application started : lst6: Directory "c:\\windows\\*.*"
```

The standard list box has two component specific attributes which relate to multiple selection. If you check the **Multiple Selection**, then a selection of an entry in the list box will remain highlighted, even when you make another selection. In order to deselect an entry, you must re-select it. The option **Extended Selection** allows the selection by also pressing the *SHIFT* and/or *CTRL* keys, to specify a range of entries.

Note:
Pressing the **SHIFT** *key and clicking on an entry will highlight all entries between the first and the subsequent selection. If you press the* **CTRL** *key and make selections, you can highlight non-contiguous ranges.*

The **Multiple Selection** setting also offers the specification of a **Delimiter for drag&drop operations**. Assume that an input field contains the entry “John;Martin;James“. If the delimiter has been set to “;“ and a drag&drop operation to a list box lst1 is performed, the list box will subsequently contain three entries (*rows*) - each entry representing one of the names. If on the other hand, a delimiter has not been defined, then the same drag&drop operation will result in one entry (*row*) containing the string “John;Martin;James“.(List-/ combo box: Attribute Drag&Drop)

An additional setting permits bitmaps to be displayed in the list box. Place a bitmap on the screen and set its attributes to allow drag&drop operations. Next, select the **Show bitmaps** option on the specific attributes of the list box. Finally, you must also drag&drop enable the list box. Go into run mode. Click on the bitmap, keep the left mouse button pressed and move it into the list box. Once you let go of the left mouse button, the bitmap will be sized and displayed in the list box. The list box in the menu option File | Preferences offers an example of what this might look like in use.

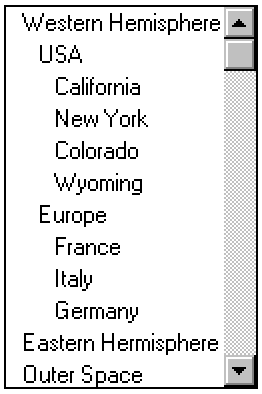
You can fill a list box either by executing a link or by directing a data source into it. Since this field is parsed, you can also declare a variable (i.e. *txt1*) and then have the value of the variable be assigned to the list box. (Section Links: Combo-/ List box)

The value of a list box corresponds to the selected entry. If no entry is selected, the value of the list box is an empty string.

Note:
This behavior can lead to the following situation. If you have set the list box to Multiple Selection and you have deselected an entry which doesn’t happen to be the first entry, and no other entries have been selected, the list box will have the last deselected entry as its value. This is due to the last deselected entry still having the focus.

A list box set to single selection will always display the selected entry as its value, or if nothing has been selected, an empty string. Selecting an entry in a single selection list box is identical to setting the focus on an entry. In a list box with multiple selection, the selection does not necessarily correspond to the focus. The focus is always on the last selection (or deselection).
If you wish to determine whether the value of the list box corresponds to a selected entry, you can employ the following trick. Drag&Drop the list box to an invisible (presumably) input field. If the length of the input field is greater than zero, the entry in the list box had the focus and was selected.

Collapsed List

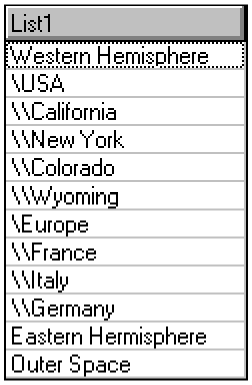

Western Hemisphere
USA
California
New York
Colorado
Wyoming
Europe
France
Italy
Germany
Eastern Hermisphere
Outer Space

This is a hierarchical list. The preceding ´+´ sign, symbolizes that entries beneath this entry exist. If you double-click on such an entry, the list will expand to show the other entries. These entries are slightly indented to make the list easier to read. Entries on this level can again represent another level. If this is the case, these entries have a preceding ´+´ sign themselves. In the above example, the entry ´New York´ belongs to the third level. The top level is ´Western Hemisphere´, the second level containing ´New York´ is entitled ´USA´. There are no practical limits to the depth of the hierarchy.

The component specific attributes permit the selection of a border and/or vertical and horizontal scroll bars for collapsed list boxes. The options **File list**, **Sorted**, **Multiple Columns**, **Show Bitmaps,** and **Multiple/ Extended Selection** have no meaning for collapsed list boxes.

Note:
*Please take note that the commands **Collapse List** and **Expand List** in the message section associated with List-/Combo boxes are only available for this special type of list box. (Section Links: Combo-/ List box).*

The collapsed list must always be filled from an external data source or the listbox specific link commands *Add string* or *Add string unique*. Entries in the data source which do not begin with a ´\´ will be displayed as top level entries. If an entry in the data source is preceded by a ´\´, it will be demoted to the second level and the entry preceding this entry in the data source will receive a ´+´ sign, to symbolize the parent status. If an entry in the data source contains ´\\´, it will be displayed at the third level. The records in the data source corresponding to the above example would have to look like this:

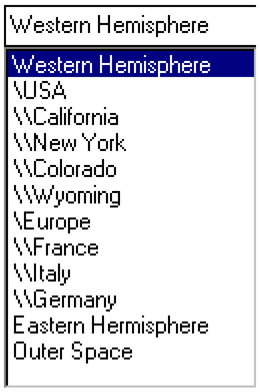

List1
Western Hemisphere
\USA
\\California
\\New York
\\Colorado
\\Wyoming
\Europe
\\France
\\Italy
\\Germany
Eastern Hermisphere
Outer Space

Entry in Data Source	**Displayed in List box**
TopLevel-A	TopLevel-A
TopLevel-B	TopLevel-B

\SecondLevel-B1	SecondLevel-B1
\\ThirdLevel-B11	ThirdLevel-B11
\\ThirdLevel-B12	ThirdLevel-B12
\SecondLevel-B2	SecondLevel-B2

Note:
The sequence of the entries is crucial for the order in this type of list box!

Combo box



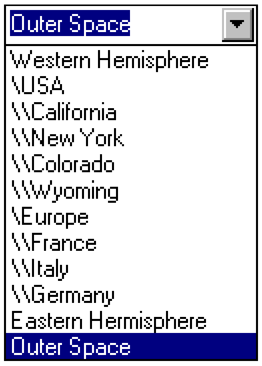
A combo box corresponds in its appearance to a combination of a list box and an input field. If you make a selection in the ´list box´ part, the selection is displayed in the ´input field´ portion. Both portions are always visible. The value of the combo box is whatever is displayed in the ´input field´ portion.

If the combo box has more entries than can be displayed, MindMap generates a vertical scroll bar. By entering the first characters of an entry, the visible part of the combo box is automatically scrolled to show entries corresponding to the character sequence. This behavior is only displayed if the combo box contains entries which correspond to the character sequence which has been input.

The component specific attribute permits the setting of a border, a vertical scroll bar and whether the entries are to be displayed in a sorted order. The options **Horizontal Scrollbar**, **Multiple Columns**, **Show Bitmaps** and **Multiple/ Extended Selection** have no meaning.

You can fill a combo box either via a link or from an external data source. (Section Links: Combo-/ List box)

Drop Down/ Drop Down List



These two types of list boxes differ only slightly. Both are displayed as an ´input field´ with a button to the right of it. Their behavior only differs in regards to user entry. A Drop Down allows the user to enter a character string in the upper part of the Drop Down. It then will position to a corresponding entry, given that one exists. A Drop Down list does not permit any user entry. If you press a key, the list box will scroll to the first entry corresponding to the character you have entered. This entry will be displayed. To make a selection the user must drop down the list and select from the displayed entries.

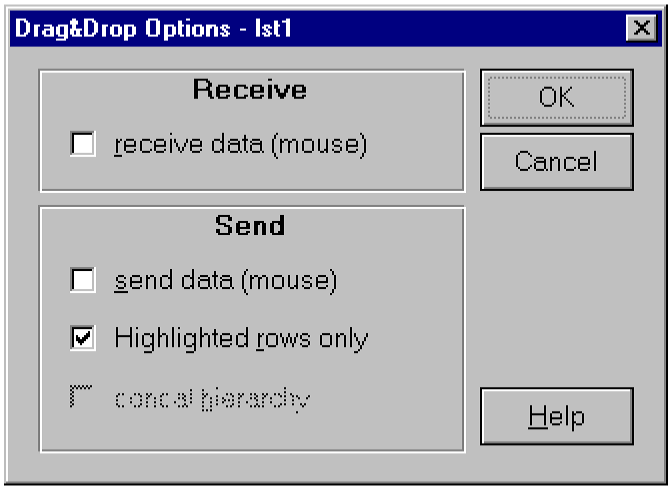
Both types of list boxes have the value corresponding to the value displayed in the upper portion of the list box.

In regards to the component specific attributes, the only difference between both types is that the Drop Down is able to display bitmaps.

Both Drop Down and Drop Down Lists can either be filled via links or from an external data source.
(Section Links: Combo-/ List box).

List-/ Combo box Component: Attribute Drag&drop

The drag&drop attribute for collapsed list boxes offers a special option. This allows either only the child or the child and (*all*) the parent(s) to be returned as the value for the list box. In the above example, not having selected the option **concatenate hierarchy** would return ´New York´ as value. If the option were selected, the value would be ´Western Hemisphere|USA|New York´.


Drag&Drop Options - lst1

Receive

[ ] receive data (mouse)

Send

[ ] send data (mouse)

[✓] Highlighted rows only

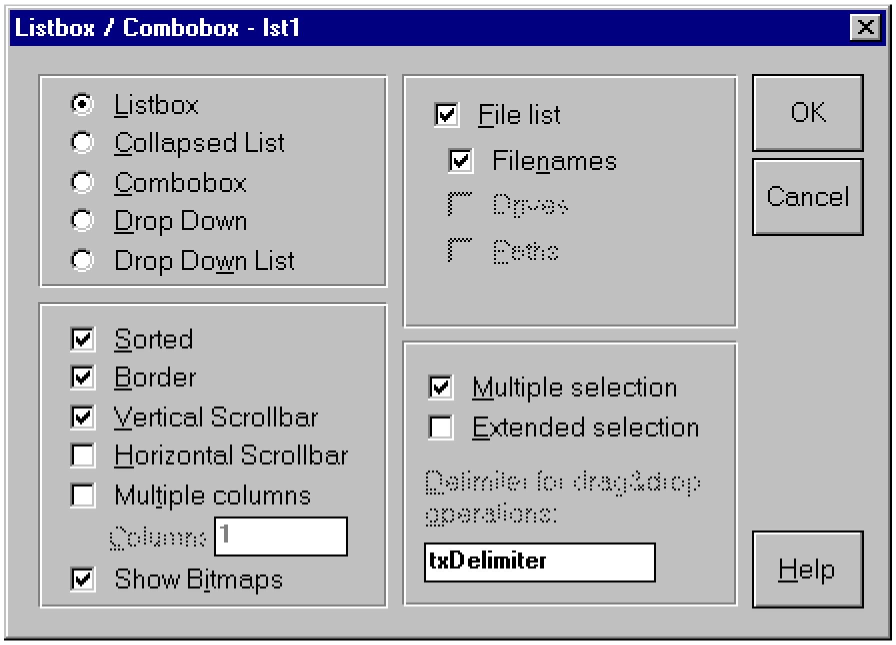
[ ] concat hierarchy

Note:
If you want to drag&drop a value from list-/ combo boxes, you must select the appropriate entry unless you uncheck the Highlighted rows only checkbox.

If you wish to drag&drop either to or from a multiple selection or extended selection list-/ combo box, you must specify the **Delimiter for drag&drop operations** in the Component Specific Attribute setting. The default setting is **crlf** (*carriage return, line feed*), which causes each entry to be displayed on its own line. If you are sending the contents of a list-/ combo box to another component, it is expected that the other component can deal with delimiters (*which is true for data tables as well as for multiple line input fields*).

It is quite common for one and the same list-/ combo box to send and receive data via drag&drop. In such cases, it might be necessary to use different delimiters depending on the direction of the data flow. In such a case, define a variable - presumably a text field - and assign to it the appropriate string (Section Links: Change Attributes)

List/ Combo boxes


Listbox / Combobox - lst1

(● Listbox) ○ Collapsed List ○ Combobox ○ Drop Down ○ Drop Down List

[✓] Sorted
[✓] Border
[✓] Vertical Scrollbar
[ ] Horizontal Scrollbar
[ ] Multiple columns
Columns: 1
[✓] Show Bitmaps

[✓] File list
[✓] Filenames
[ ] Drives
[ ] Paths

[✓] Multiple selection
[ ] Extended selection
Delimiter for drag&drop operations: txDelimiter

OK
Cancel
Help

List-/ Combo box Component: Additional Events

The following table shows the additional events in the links dialog box for list boxes.

Event	**Description**
Cursor changed	Whenever the user clicks on an entry in the list box, this event will be generated.
Double-click	Whenever the user double-clicks on an entry in the list box, this event will be generated.
Receive focus	Receive focus performs a message when the user places the cursor into the list box or tabs to it.
Lose focus	Lose focus causes an event when the user places the cursor out of the list box; i.e., when the user clicks on another component or when the focus is directed to a different component by pressing the TAB key.
Edit Change	This event is for combo boxes only. It is generated if you edit the content of the header row of the combo box.
Drop down	Whenever the user opens the Drop Down or Drop Down List, this event is generated.
File select	This option is only available in conjunction with the component specific attribute, File list. When a file is selected, this event is generated.

Special

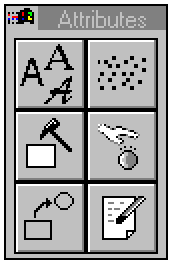
Please note that List/ Combo boxes offer special parser functions. (Parser)

Data Table Component

This component is used to display data in a table format, a bit similar to a spread sheet. The data can either originate from an external data source or can be input via the keyboard. An additional method to populate a data table is by assigning values to it via the parser function SetDataTable (*Parser*).

In order to place such a component, click on the icon on the toolbox or select from the **Objects | Data Table** menu. Again the cursor will change its appearance to reflect the selection. It will appear as a crosshair with a little representation of the data table. Move the cursor to the position you wish to place the data table. Then press the left mouse button and drag the mouse to reflect the size you desire to have for the data table. As soon as you release the mouse button, the data table will become visible. You are then prompted to enter a name for the data table. Either accept the default suggestion or type in a specific name.

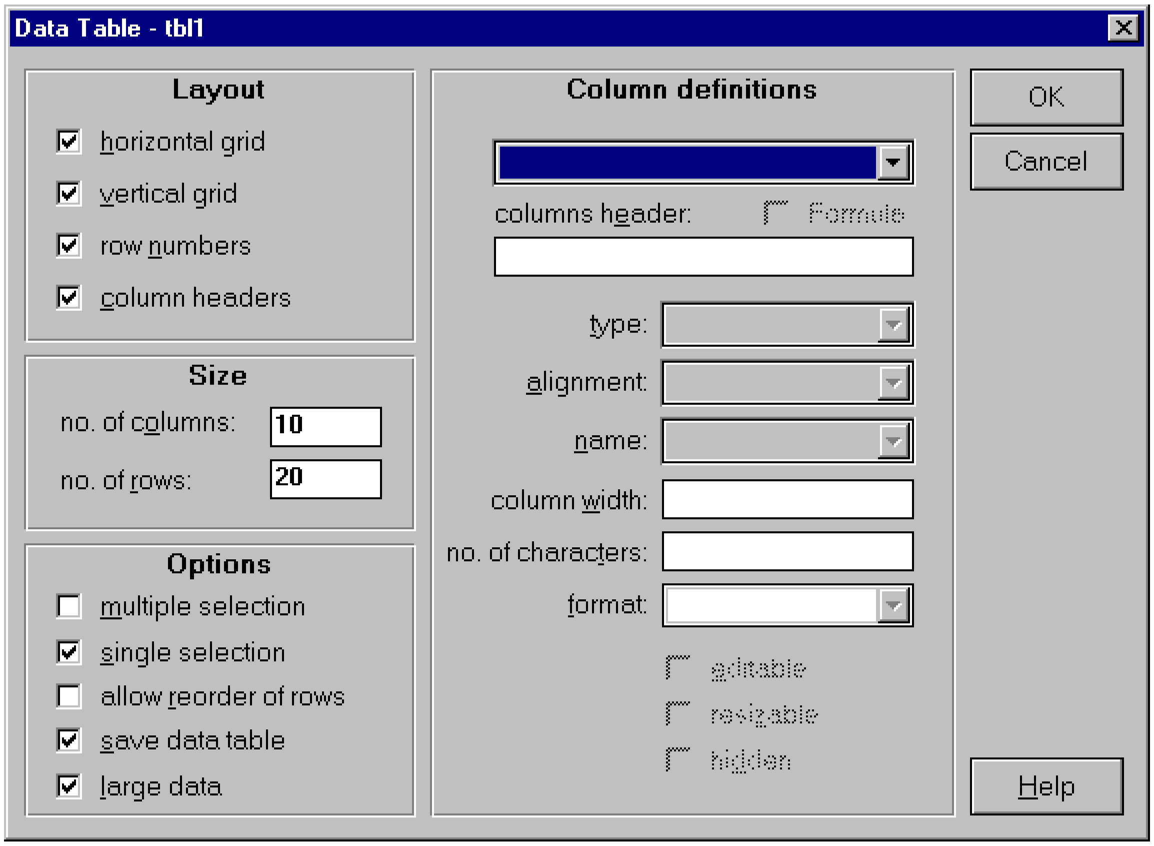
Data Table Component: Attributes



The **Font** attribute permits the setting of the common font definitions.

The common attributes are further explained in Format Menu

Data Table Component: Component Specific Attributes


Data Table - tbl1

Layout
- [✓] horizontal grid
- [✓] vertical grid
- [✓] row numbers
- [✓] column headers

Size
no. of columns: 10
no. of rows: 20

Options
- [ ] multiple selection
- [✓] single selection
- [ ] allow reorder of rows
- [✓] save data table
- [✓] large data

Column definitions
columns header: [ ] Formula
type:
alignment:
name:
column width:
no. of characters:
format:
- [ ] editable
- [ ] resizable
- [ ] hidden

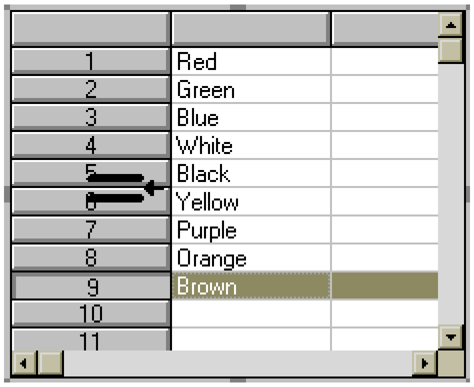
Layout: If necessary, you can decide to have the **row numbers** display. This might be of assistance when your data table contains many entries. Displaying **row numbers** is mandatory if you wish to permit rearrangement of rows at runtime. (*allow reorder of rows*)

The option **Column Headers** allows you to enter column headers, given that you have defined such in the **Column definitions** option. You can assign headers that are different from the field's name in the database, assuming that you have connected the data table to a data source.

Size: The layout size function specifies the number of **columns** and **rows**. Here you can determine the exact number of rows and/or columns. The selection you make for the number of rows will not be affected at runtime. The setting of the rows will be overridden, if you have connected the data table to a data source and the data set to be displayed contains more rows (= *records*).

Options: In this section, one may set whether this data table will allow **single** or **multiple selection**. The multiple selection highlights every row you click on or a series of consecutive rows if you keep the left mouse button pressed, whereas the single selection highlights the selected row only (*and deselects any other rows previously selected*), while every new mouse-click selects another row.

It is sometimes necessary to set the option **allow the reorder of rows.** This is only possible for data tables with **single selection**. This option permits the user, at run time, to select a row and move it to another row in the same data table. When performing this operation, moving the cursor above a different row in the data table will make the cursor change to show two horizontal lines with an arrow pointing between them. This is to symbolize inserting the selected one between two existing rows. Once you click on the left mouse button again, assuming the horizontal line cursor is being displayed, it will move the row to the new position. Please remember that it is necessary to have the row numbers display, in order to perform the reordering thereof.



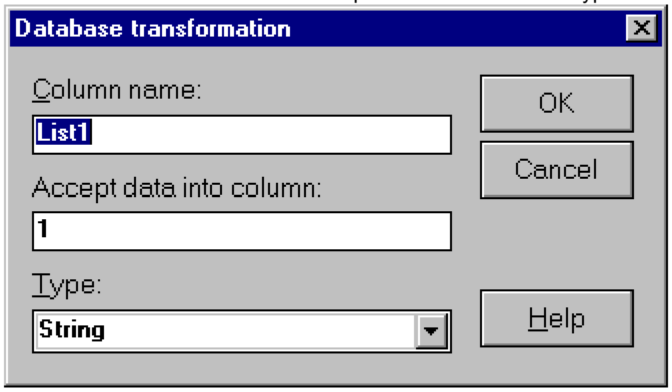
Note:
Please note that when you are rearranging rows in a data table, which is connected to a data source, only the row's order in the data table is affected. The order of the records, as they are stored in the external data source, is not affected.

In most cases, the data table is used to display data coming from an external data source. Thus the contents of the data table is constantly changing, depending on the data stream coming in. In some cases though, it is desirable to have a static set of data represented in the data table. In this case, assign desired contents to the data table and select the **save data table** option. The contents of the data table will then be saved into the application file.

Column definitions: At the top right corner of the dialog box you have the option of defining specific names for each column. Drop down the list to select a column heading. A column is automatically assigned a header consisting of the string ´Column´ and an increment. To change it, select the appropriate header and type in the desired name in the field immediately below.

Columns Header: Normally you assign a name to a column. If you want to specify the names of the columns, you have to select the appropriate column out of the drop down list that is placed below the **column definitions** heading. The cursor jumps to the input field below and you can type in the column header. This can be done for all columns. You also can assign a **Formula** as a column header. This means that the entry will be interpreted by the parser. This allows you to change the column heading dynamically at runtime. This feature can come in handy when developing multilingual applications.

In the case that you are assigning an external data source to the data table, you are prompted, during the assignment, to define a column heading. If you opt to use the default heading, it will be constructed out of the string used as field name in the database. This assignment dialog also permits you to define to which column in the data table you wish to direct the data source. You are also offered an option to define the data type. In all three cases, you can choose to keep the default settings.



Column name: List1

Accept data into column: 1

Type: String

Type: In this section, you select a special type out of the following: String, Float, Integer, Date/Time. The type setting influences the **format** list. In case the data table is connected to a data source, the data type of the column in the data source is automatically converted to a reasonable type out of the four available data table types. Please check that this type conversion meets the needs of your application.

Alignment: Each column can be formatted with the alignment function. Use this option to set the alignment of the entries in the data table. You can choose from left, right or center aligned.

Name: The alignment of the heading is independent of the alignment of the actual data entries. The header of the column can be aligned left, right or center.

Column width: This setting refers to the visible width of a column. This setting is independent of the option **no. of characters**. Another way to define the column width can be set with the mouse. If you position the mouse cursor on a header and move to the edge of the header the cursor will change to a cross with arrows on the left and right side. Press the left mouse button and pull the column width to the desired size. This new size will be shown in the column definition section, when you access the Component Specific Attribute dialog box the next time.

If you connect a data source to the data table, the database system will automatically change the column width to reflect the widths of the appropriate database columns.

No. of characters: The number of characters for each column can be specified here. MindMap sets the number of characters to 80 by default. Depending on the setting of **column width** the visible portion could be larger. The column width must not exceed 1024 characters.

Format: Depending on the **type** of column different formats are possible. The entries in the list are merely meant as examples. You can alter these at anytime.

Column type	**Possible Formats**	**Description**
String	@@@@	Four alpha-numerical characters can be entered.
	Aaaaaaaa	Eight alpha-numerical characters or punctuation marks can be entered.
	nnn-nnn-nnnn	Ten alpha-numerical characters or punctuation marks can be entered. MindMap automatically supplies the hyphenations at the specified positions.
Float	#,##0.00	The number is represented with two decimal digits. The thousands are separated with a comma. Example: 34,112.32
	#.##	The number is represented with two decimal digits. Example: 34112.32
	+#.####	The number is preceded with a plus sign and has four decimal digits.
Integer	#.##0.00	If a number with decimal digits is entered, these are truncated. The thousands are separated with a comma. Example: 23,345.00
	#.##	If a number with decimal digits is entered, these are truncated. Example: 3.78 becomes 3.00
	+#.####	A number with four decimal digits will be preceded with a plus sign. Example: +2341.0000

Date/Time	DDD MM/DD/YYYY	The date is displayed with a Day of Week abbreviation. The month is represented as a two-digit number. The day is also represented with two digits, the year will be displayed in four digits. Example: Sun 12/03/1995
	DDDD MM/DD/YYYY	In this case the date will be displayed in the same format as in the example above, with the exception that the weekday is not abbreviated. Example: Sunday 12/03/1995
	MM/DD/YY	In this example the weekday is not displayed at all. Example: 12/03/95
	MM/DD/YY HH:MM	The date again is displayed as described in the above examples. The time is appended. It is displayed military style (“00“ through “23“ hours) and 2 digit minutes. Example: 12/03/95 21:14
	MM/DD/YY hh:mm P	The date is displayed as in the above example, except that the US style of displaying hours in two cycles of “00“ to “12“ followed by either AM or PM. Example: 12/03/95 09:14 PM

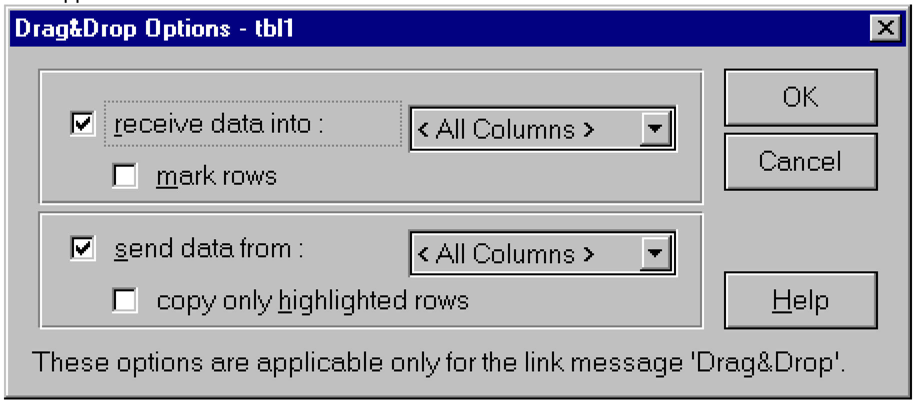
You can adjust the rows to be set to **editable**, **resizable** or **hidden**.

In the case of **editable,** the user can change the contents of a cell. If you intend to use a data table only to present data (*and not allow the user to change the values*), you should switch off the option **editable.**

Resizable allows the user to change the column width with the mouse. You can set the column to **hidden** so that it is not seen by the user. This feature is very important in conjunction with a data source. You can connect hidden columns in the data table to relevant fields in the data source. Unique keys, numerical sort order indices, etc., can thus be kept synchronized, without having to display them.

Data Table Component: Attribute Drag&drop

The Drag&drop dialog box exclusively controls the behavior of the data table when a drag&drop link message is applied. Once **the receive data into** and/or **send data from** checkboxes are checked, it is possible to move data into and from a single cell, by dragging with the mouse. Please note that dragging information between cells within the same data table is not supported.



☑ receive data into : < All Columns >
☐ mark rows
☑ send data from : < All Columns >
☐ copy only highlighted rows

These options are applicable only for the link message 'Drag&Drop'.

This dialog box permits you to specify whether the data table is to receive data via a drag&drop link message. You can specify that data is to be received with the **receive data into** option. If you have made this selection, you can determine whether the data is to be received in all columns (**All columns**) or restrict it to individual columns. If you perform the drag&drop link message from a list box or from a multiline input field into a single column of the data table, all rows that match the lines of the drag&drop source will be highlighted. Obviously, this operation only works if the **multiple selection** check box is checked in the component specific attributes dialog.

Note:
Again, please be aware of the fact that selecting a column for sending or receiving drag&drop information only applies to the drag&drop link message. It does not affect the ability to drag&drop information using the mouse.

If you intend to permit data to be sent, you must select the option **send data from**. Then specify whether all columns or only the marked columns are to be sent. **Copy only highlighted rows** allows only selected rows to be sent.

Note:
When sending drag & drop data, the data table component will convert its contents in a way that cells within the same row will be separated by the TAB character and rows are separated with the CRLF (carriage return/line feed) character sequence.
If drag & drop data is sent to the data table the various cells will be filled with the information from the drag & drop source component if they are formatted in the same way.

Data Table Component: Additional Events

Event	Description
Cursor changed	Whenever the user clicks on an entry in the list box, this event will be generated.
Double-click	Whenever the user double-clicks on an entry in the list box, this event will be generated.
Receive focus	Receive focus performs a message when the user places the cursor into the list box or tabs to it.
Lose focus	Lose focus causes an event when the user places the cursor out of the list box; i.e., when he clicks on another component or when the focus is directed to a different component by pressing the *TAB* key.
Abort edit mode	If you have selected a cell in the data table and press the Escape key, this event is generated.
Begin edit mode	This event is generated when you begin to input data into a cell.
Cell changed	This event is called as soon as a cell has been changed.
Order changed	The change of the data table's order generates this event. This event is only defined for data tables with the Component Specific Attribute **allow reorder of rows** set.

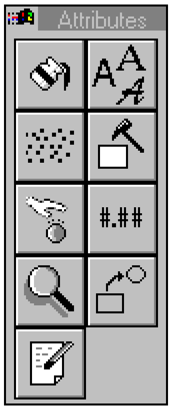
Input Field Component

Input fields are generally used to either get input from the user at runtime or to display data originating somewhere else.

In order to place an input field, click on the icon on the toolbox or make the selection via the menu option. Once you have selected the input field, the cursor will change its appearance and will become a crosshair with a little rectangle attached to symbolize the input field draw mode. Move the mouse to the screen position where you wish to place the input field. Press the left mouse button, drag the mouse towards the lower right screen and release the mouse button. You are immediately prompted for the component name. You can either choose to use the default name or simply type in the new name.

Depending on the font selection, opening an input field over a certain height will produce vertical scroll bars. Vertical scroll bars signal that the input field can display multiple lines.

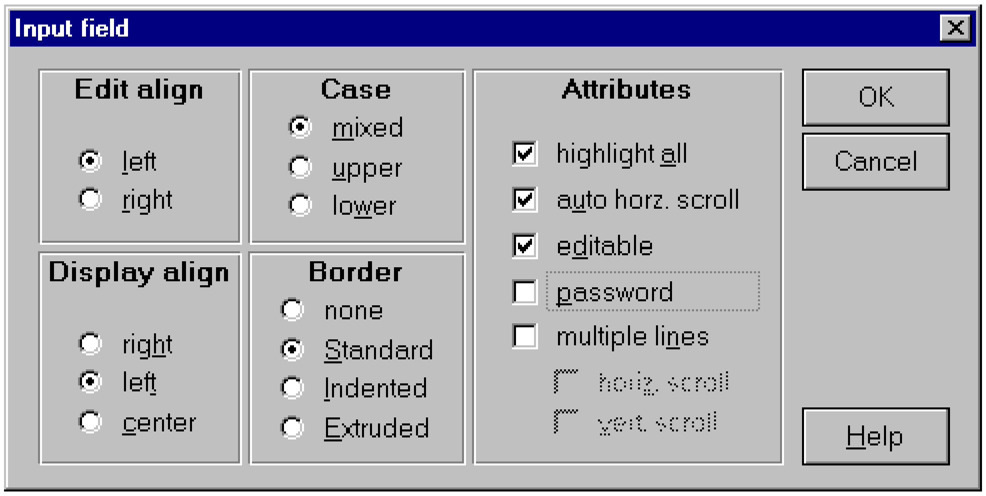
Input Field Component: Attributes



The **Color** attribute defines the background color for input fields. The **Font** icon permits the standard font settings, as well as the color definition for the font.

For additional information regarding formatting Format Menu

Input Field Component: Component Specific Attributes



Input field

Edit align: [✓ left] right

Display align: right [✓ left] center

Case: [✓ mixed] upper lower

Border: none [✓ Standard] Indented Extruded

Attributes: [✓ highlight all] [✓ auto horz. scroll] [✓ editable] password multiple lines horiz. scroll vert. scroll

The **Edit align** attribute enables you to define were the text should be aligned when the cursor is placed in the input field. The options are **left** or **right**.

Independent of the selection you chose for alignment when editing, the setting of **Display align** will take effect once the input field loses the focus. The Display align option allows **right**, **left**, or **center** alignment.

The attribute **Case** permits you to determine how the input field will deal with the user input in regard to capitalization. If you select the setting **mixed** (*default setting*), the input will be displayed in exactly the same manner the user inputs it. The other two alternatives are **upper** and **lower**. In the case of **upper**, all input will be displayed in capital letters, regardless of how the user inputs the characters. If you have chosen **lower**, then all characters will be displayed in lower case, again regardless of how they were input. Please note that this setting also affects how the content of the input field is passed to other components, especially database components.

The **Border** option allows you to choose between various settings, which merely affect the appearance of the input field. These settings are specific to the input field, so they can be found under the component specific attribute.

The **highlight all** option will select all characters in the input field, once the input field receives the focus. If the user begins to type, the selected characters are deleted
(*overwrite mode*). If you do not check this option, then the cursor will be set to the left of the first character in the input field, once the input field receives the focus (*insert mode*).

Sometimes there may not be enough space for long input fields on your page. If you do have long strings, you can set the Input field to **automatic horizontal scroll**. This allows you to have a small input field and the user can scroll through the string to view it completely.

Note:
You will not have scroll bars with scrolling input fields, unless they are set to have multiple lines. The user must position the cursor in the input field and move to the end to see the whole string.

A better way is to use **multiple lines** for those input fields with long strings. If you choose this option, the **edit align**, **display align** and **case** section are grayed out and they are not accessible. If you decided to use **multiple line** Input fields, you may select to have **horizontal** and/or **vertical scroll bars** available.

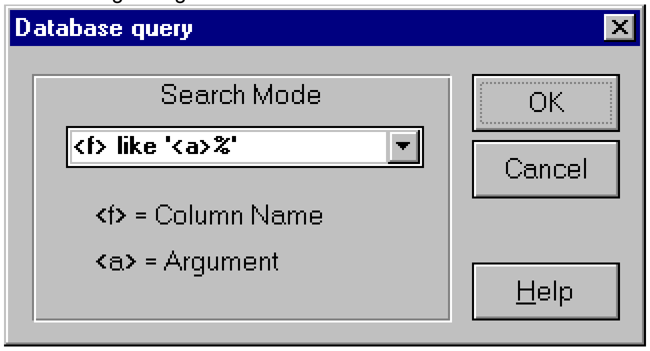
Input fields can be set to **editable** or not.

A **password** function is integrated, as well. This function shows only asterisks in the input field and the user's input is safe and hidden. However, if you assign the value of the input field to another component, the actual password (*not the asterisks*) will be transferred.

Database Query

You can perform a database search on the content of an Input field that is connected to the database. (See section Database Manager Component) . The search mode you defined here will be used.

Note:
*These search mode settings are valid for the commands **Search for fields**. If you dynamically construct your own SQL strings at runtime using the command **SQL Command,** you do not need to consider these settings.*
The attributes toolbox of input fields offers an icon for database queries. If you click on the icon, you will be offered the following dialog box:



Search Mode

<f> like '<a>%'

<f> = Column Name

<a> = Argument

If you open the list, you can select from a predefined list of various **Search Modes**:

Search Mode	**Description**
<f> = '<a>'	The first mode searches for a stringvalue that equals the argument. If the content of the input field <a> is 'test', then the result set will only consist of strings which match this string exactly. Please note that this option is also applicable for searching other data types than strings, such as numeric values, dates and times and boolean data.
<f> like '<a>%'	The second mode searches for a string that starts like the argument. This condition is satisfied with such results as 'test', 'tests', 'tested', 'tester', etc. This option is applicable only for string searches.
<f> like '%<a>%'	The last mode searches for a part of a string that is like the argument.

Achievable results include 'test', 'tested' and 'pre-tested', etc.

Again, this option is applicable only for string searches.

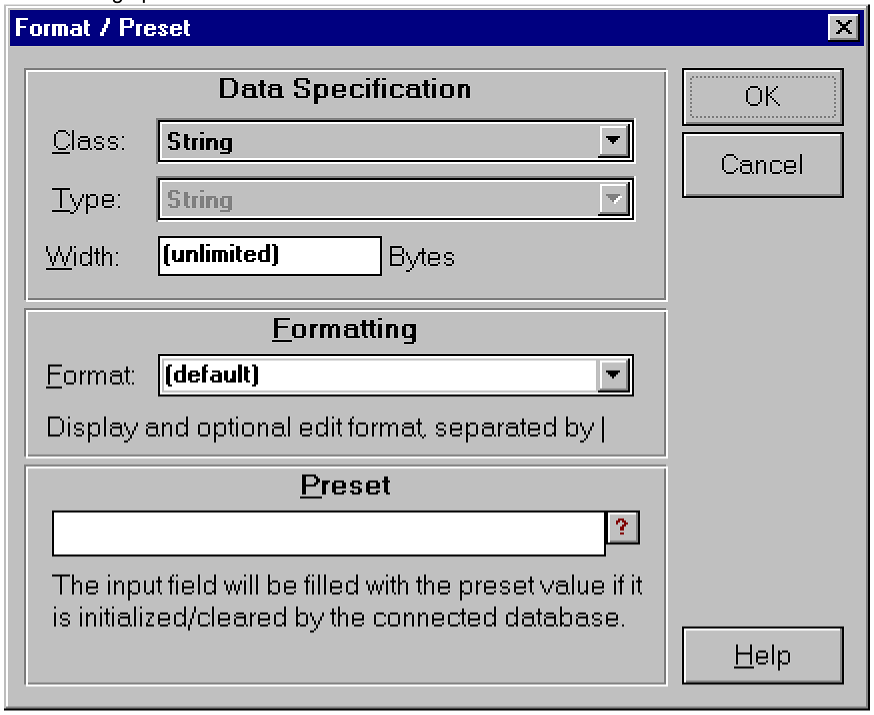
"<f>" refers to column name, where the field name in the data source is meant; "<a>" is the argument, or, in other words, the contents of the input field.

Note:
You may have noticed that, besides selecting one of the predefined values, any text can be entered in the search mode combo box.
If you notice   that a specific database driver supports search conditions other than '=' or 'like', you may enter these verbs in this combo box.
For example, try using comparison operators such as <, <=, >, >= and <>.
Also note that the contents of this combo box is evaluated by the parser at run time. This means that you are allowed to enter the name of another component, as long as it contains a syntactically correct comparison operator with respect to the database the input field is connected to.

Format/ Preset

#.##

Input fields have various formatting options. If you click on this button on the attribute toolbox, you will be presented with the following options:



Data Specification

Class: String

Type: String

Width: (unlimited) Bytes

Formatting

Format: (default)

Display and optional edit format, separated by |

Preset

The input field will be filled with the preset value if it is initialized/cleared by the connected database.

If you select the entry **Number** from the **Class** list, you will be offered different **Types**. The **Type** list contains all data types relating to the selected **Class**. If only one type is applicable for the selected class, this entry shows the appropriate setting and is disabled.

The **Width** field allows you to specify how many characters may be entered into the input field- it is only defined for the data type string. The default setting for the **Width** field in this case is unlimited. Unlimited in this sense means that a limitation depending on available internal system resources applies. Expect this limitation to be around 16000 characters for single line input fields and 32000 characters for multiple line input fields.

Note:
*If you connect a data source component with an input field, the input field picks up its settings for **Class**, **Type** and **Width** automatically and thus corresponds to the field in the data source.*
This conversion is accomplished by using the information that a specific data source returns. Some data sources may supply this information incomplete or not recognizable for the input field. MindMap will attempt to match these types as closely as possible. However, you should always verify, that a reasonable type has been selected.

If you select **Format**, assuming the data type supports it, you can choose a format from the list or define your own, according to the masks listed below.

Class	Token	Replacement
Boole		

	<verb1>;	The first verb will be displayed if the boolean value is **true**, whereas the second verb will be displayed if the value is **false**.
	<verb1>	Assigning a numerical value of zero to a boolean input field will set to false, all other values will set to true.
Mask		
	#	Any numeric digit (0-9)
	@	Any alphabetic character (a-z, A-Z)
	!	Any punctuation character
	*	Any single printable character
	\	Causes the next character to be literal
		Any other character will be displayed as is.
Number Currency		If there are two formats separated by ; MindMap will use the second format to display negative values.
	#	Any numeric digit (0-9)
	.	Decimal point delimiter
	,	Separating delimiter for thousands
	+ or -	plus or minus sign
	\	Causes the next character to be literal
		Any other character will be displayed as is.
Date		The following tokens can be concatenated to create a date format using any reasonable delimiting characters as can be seen from the following examples:
		mm/dd/yy 08/28/98
		mm/dd/yyyy 08/28/1998
		Dddd, Mmm dd, yyyy Friday, Aug 28, 1998
	m	Shows the month. (1..12)
	mm	Shows the month using two digits with preceding 0. (01..12)
	d	Shows the day. (1..31)

	dd	Shows the day using two digits with preceding 0. (01..31)
	Ddd	Shows the day-of-week abbreviated to 3 digits. (Sun..Sat)
	Dddd	Shows the day-of-week. (Sunday..Saturday)
	Mmm	Shows the name of the month abbreviated to 3 digits. (Jan..Dec)
	Mmmm	Shows the name of the month. (January..December)
	yy	Shows the year using two digits.
	yyyy	Shows the year using four digits (therefore including the century).
Time		The following tokens can be concatenated to create a date format using any reasonable delimiting characters.
	h	Shows the hour. (0..23)
	hh	Shows the hour using two digits with preceding 0. (00..23)
	m	Shows the minute. (0..59)
	mm	Shows the minute using two digits with preceding 0. (00..59)
	s	Shows the second. (0..59)
	ss	Shows the second using two digits with preceding 0. (00..59)
Date/Time		This format allows a concatenation of the tokens described for Date and Time.

Please note that you may specify two formats like this separated by the | character. If you do so, the second format will be used if the input field has the focus (is in an editable state). This is useful especially for date values containing the day-of-week, since this value depends on the date settings and may not be specified independently. If no second format is given, MindMap attempts to use the same format for both edit and display mode.

The **Preset** option allows you to preset the contents of an input field, if and only if, it has been connected to a data source. If you issue the database command **Clear display fields** and have set the option **Initialize** on, the contents of the field will be cleared. If the **Preset** option has been selected, the associated value will be displayed. Please note that this value is evaluated by the parser (See section Database Manager Component)

Database Query: Additional Events

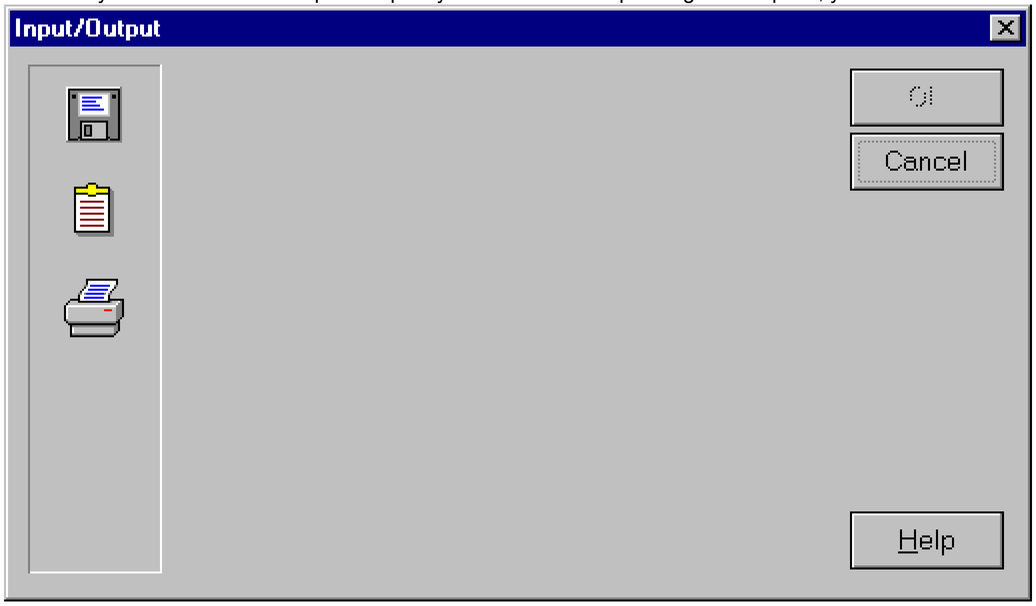
The create and edit links dialog box for input fields has some of the standard events. The few additional events are shown in the following table:

Event	Description
Enter	This event is generated when the cursor is positioned in an input field and the user pressed the ENTER-key.
Keyboard input	This event is generated when the cursor is positioned in the input field and the user presses any key.
Receive focus	This event is generated when the component receives the focus; for example, if you select it.
Lose focus	This event is generated, as soon as the input field loses the focus.

Input / Output Component

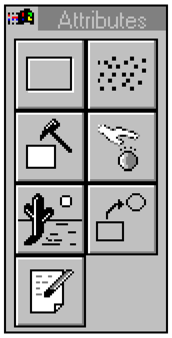
Via the File and Clipboard interface, you can import and export data (*Drag & Drop*). The Printer component permits you to output information to the printer, also via drag & drop.

Once you have clicked the Input/ Output symbol or the corresponding menu option, you will see the following dialog box:



Here you can select the appropriate symbol for **File**, **Clipboard** or **Printer**. Once you have selected one of the three options, the dialog box will be updated to reflect the resulting options. The resulting dialog box corresponds directly to the one you would see, had you selected the component specific attributes. Once you have clicked the OK button, the cursor will change its appearance to look like a crosshair with a little attached input/output component symbol. Move the cursor to the desired position on the page and drop the component. The component will be created in its standard size. Obviously, you can also drag open the component (*left mouse button pressed while dragging*). In this case, the size of the bitmap will always remain the same size and only the surrounding frame will become larger.

Input / Output Component: Attributes

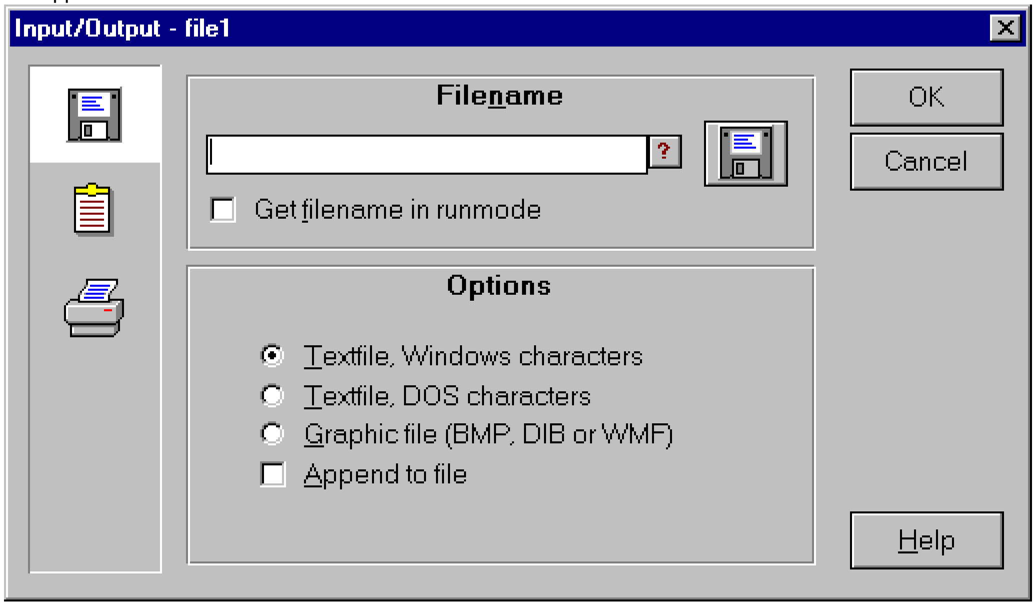


The default setting for the **Drag & drop** attribute is **send data** as well as **receive data**. The attribute **Graphic Import** offers the option of replacing the bitmap of the placed component with one of your own choice.

For further information concerning common attribute Format Menu

File

This component is designed for reading data from or writing data into a file. When you select the symbol, the dialog box will appear as follows:


Input/Output - file1

Filename

[ ] Get filename in runmode

Options

[● Textfile, Windows characters]
○ Textfile, DOS characters
○ Graphic file (BMP, DIB or WMF)
[ ] Append to file

This dialog is identical to the one which appears when you select the component specific attributes.

In this dialog box, you can input a **File name** with directory and path (*don't forget to use two backslashes to identify the directory name*) or select a file via the button with the diskette symbol. The entry for the file name is parsed so that you may also enter a component name containing the desired file name. (Parser)

If you select the option **get file name in run mode**, you do not need to make an entry for **File name**. At runtime, a dialog box will appear as soon as the drag & drop link is executed. Here the user can enter a path and file name.

In the **Options** section, either a **MS-Windows** or **DOS** characters text file can be defined. A **Graphic file** (*BMP, DIB or WMF*) could also be used. The text in the Input field could be **appended** to the text file by checking the appropriate box.

Left mouse button released: Drag & Drop file1 -> edt1

These components communicate via the drag & drop operation. In principle there are two different ways to execute the drag & drop operation. (Properties | Drag & Drop | Options). Independent of how the drag & drop option has been set, you can include this link. Such a link could look like this:

The above link would display the contents of the file specified in file1 in the input field edt1, once the user has pressed the button. If you exchange the direction of the link, then the contents of the input field edt1 would be stored in a file specified in the output file1.

Note:
In order for this to function, it is not necessary to have the file component displayed on the screen.

The other method to execute a drag & drop operation is by means of the mouse. In this case, it is not necessary to create a link. You merely set the attributes of the sending and receiving components so that both can participate in the operation. You must set the receiving component's drag & drop option to **receive data**. The sending component (*or rather the component from which the data is originating*) must be set to **send data**.

To receive the same results as in the above example (*display the contents of a text file in the input field*), at run time, move the mouse over the file component. Then press the left mouse button and drag the mouse to the input field. Once the cursor is above the input field, release the left mouse button. As a result, the contents of the file will appear in the input field. If you have chosen to **get file name in run mode,** a dialog box will appear, offering the opportunity to select a file. Once the file has been selected, its contents will appear in the input field.

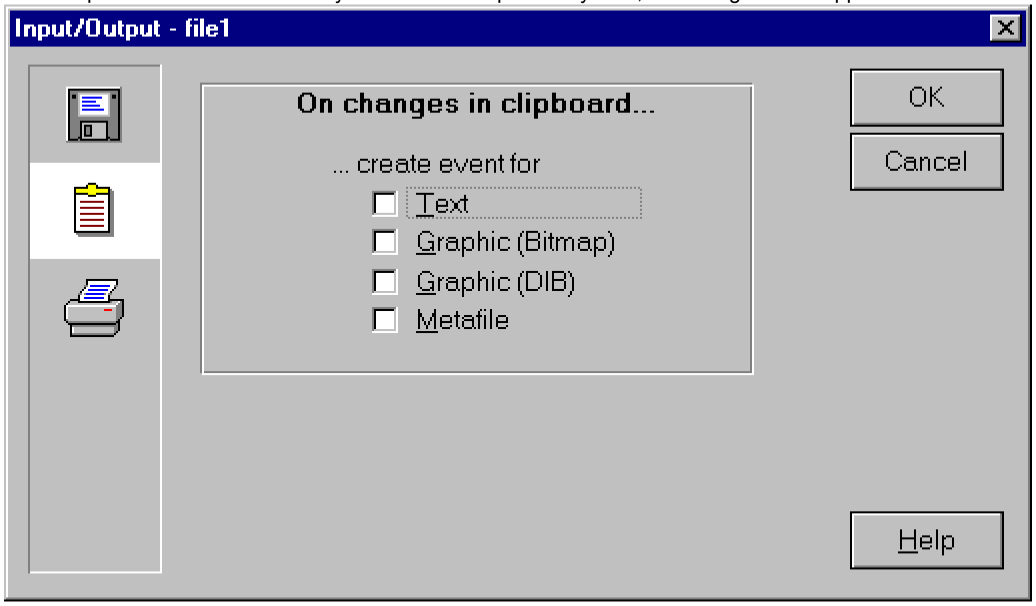
Note:
In this case, the file component must be visible on the desktop at run time.

Input / Output Component | File: Special Note

MindMap offers the facility of storing the contents of a data table in a file and, vice versa, of displaying the contents of a file in a data table. The data in the columns is separated by a TAB. The beginning of a new row is defined by a CRLF (*Carriage return line feed*).

Clipboard

The Input/Output clipboard component is similar to the file component. You can send data from the MS-Windows clipboard to an input field and vice versa. If you select the Clipboard symbol, the dialog box will appear as follows:



On changes in the clipboard, MindMap creates events for text or graphics (*BMP, DIB or WMF*).

Input / Output Component: Component Specific Attributes

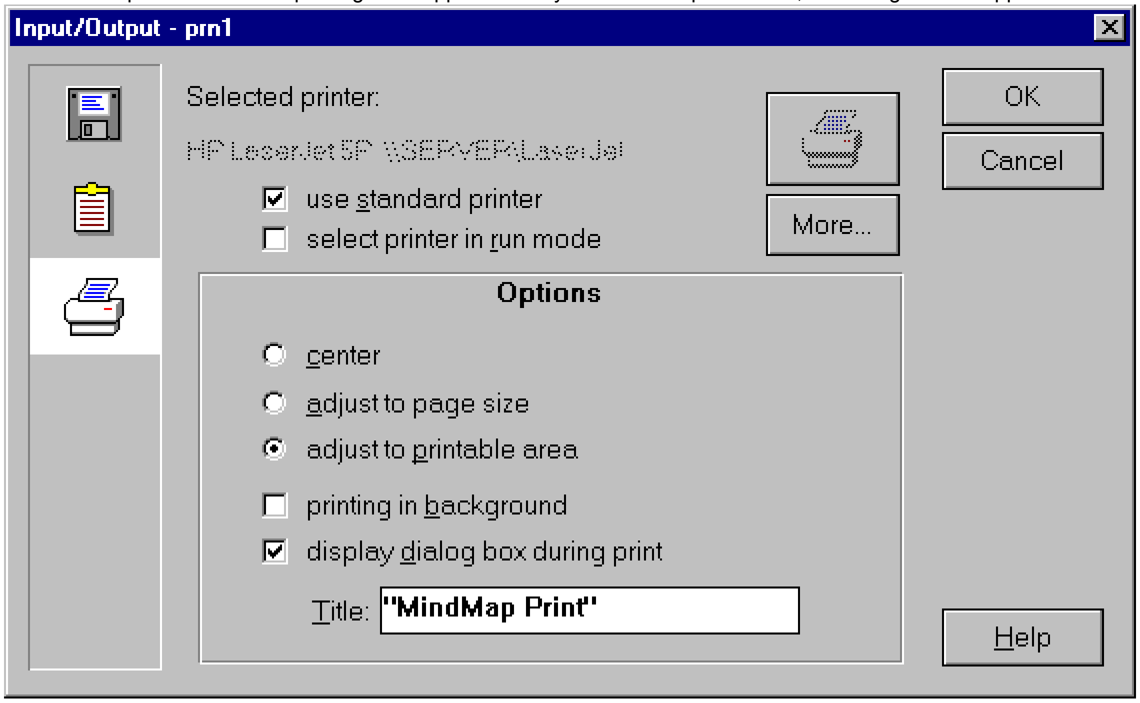
This has identical functions to the file component.
(File Component)

Input / Output Component: Additional Events

Event	Description
New data in clipboard	Whenever the MS-Windows clipboard is filled by another application, this event is generated.
Clipboard empty	This event is generated when the clipboard is emptied by another application.

Printer

This component allows for printing in an application. If you select the printer icon, the dialog box will appear as:



Selected printer:
HP LaserJet 5P \\SERVER\LaserJet

[✓] use standard printer
[ ] select printer in run mode

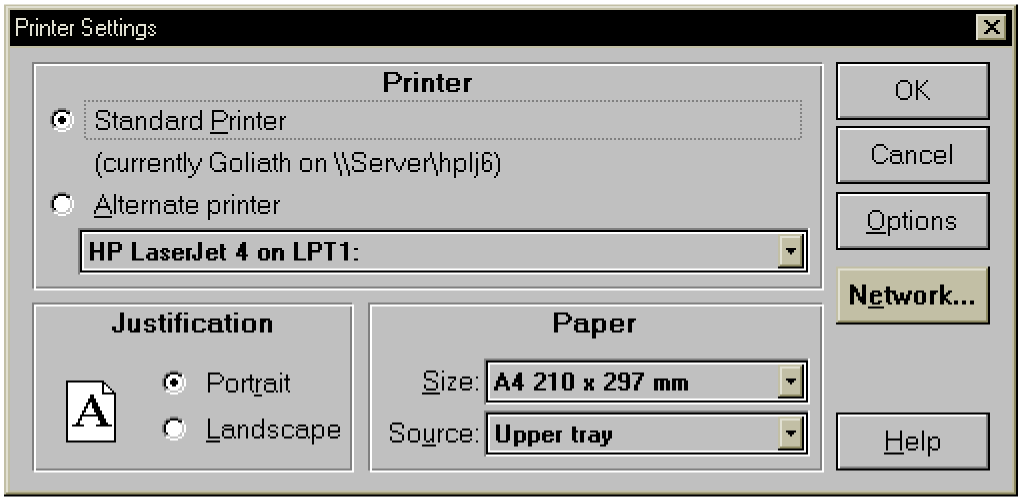
Options
( ) center
( ) adjust to page size
(●) adjust to printable area
[ ] printing in background
[✓] display dialog box during print
Title: "MindMap Print"

In the first section, MindMap shows the selected printer. By default, this is your standard MS-Windows printer; i.e., the standard printer of the system on which you are constructing the application.

You can choose whether the installed **standard printer** should be used for printing, or if the user has to **select a printer in Run mode**. The setting **standard printer** will allow for printing from the application on any computer, if a standard printer has been defined in the control panel. If you uncheck both check boxes, MindMap will print to the printer displayed as selected printer regardless of which printer has been selected as default printer.

Note:
If you move such an application to another computer please make sure that exactly the same printer on the same port is available and functioning.

If you click on the button with the **printer icon,** the printer setup dialog box pops up.


Printer Settings

Printer

Standard Printer [✓]

(currently Goliath on \\Server\hplj6)

Alternate printer

HP LaserJet 4 on LPT1:

Justification

Portrait [✓] Landscape

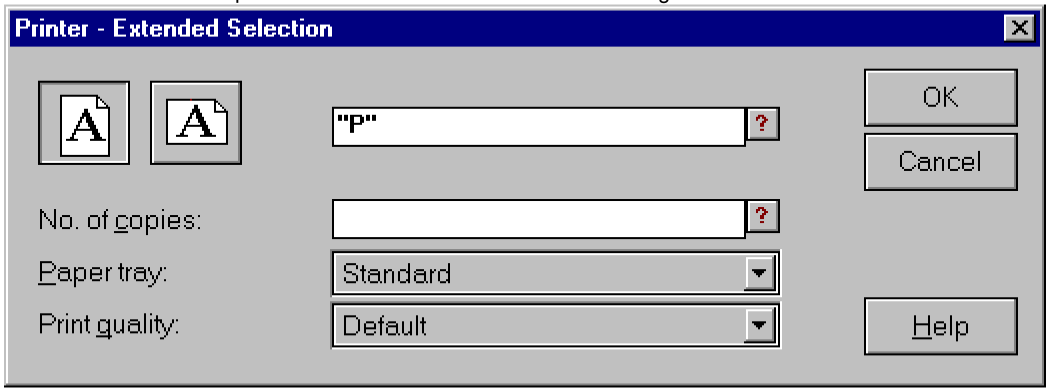
Paper

Size: A4 210 x 297 mm

Source: Upper tray

Here you can select a **Standard Printer** or an **Other Printer,** provided you have installed other printers in MS-Windows (*via Control Panel*). The **Orientation** - portrait or landscape - can be selected as well (Button **More**). The **Paper Size** and **Source** can also be changed. These specifications depend on the selected printer and are not described in detail here. The **Options** button opens your special printer setting where you may specify the print quality, for example.

The **More...** button opens the **Printer - Extended Selection** dialog box.


Printer - Extended Selection

"P"

No. of copies:

Paper tray: Standard

Print quality: Default

The **Orientation**, **Number of copies**, the **Paper tray** and the **Print quality** can be selected in this dialog box. These settings differ from those in the **Printer Setup,** in that they are parsed at run time. This allows for the entry of variables (*names of components which refer to names*). (Parser)

The section **Options** allows additional settings.

You can select **center**, **adjust to page size** or **adjust to printable area,** where **adjust to page size** is the default setting. The **center** setting will print the text or the graphic, center aligned. If you choose the setting **adjust to page size,** the printout will always be the same, regardless of the printer selected. Some printers have a wider non-printing margin. This would normally truncate the output. This can be avoided, if you choose the setting **adjust to printable area**. In this case, you select the printer and, if necessary, the output will be adjusted in size to fit the permissible area.

If you have selected **printing in background,** you can print in background while continuing to use the application in foreground. You can select the **display a dialog box during print** option, which will display a dialog box when the printing begins. The **Title** states the caption for the printer dialog box. The name of your application or the name of the printout can be stated here, for example. Since this field is parsed, you can also include component names which reference text strings. (Parser)

Input / Output Component | Printer: Special Note

A printer will only permit output. In addition, you have the two options mentioned above, if you wish to initiate an output. (File Component).

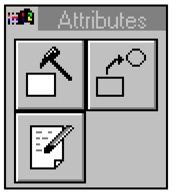
Note:
Normally the contents of the output page component is printed. This implies that you place other components (data tables, graphics, etc.) on the output page component. Then you drag & drop the output page to the printer component.

You can also print graphic components directly, without going through the output page component. In this case, you just drag & drop the graphic component directly to the printer component.

Menu Component

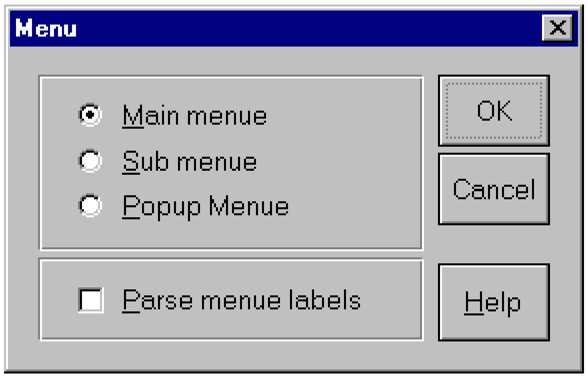
You can create menu items for MindMap applications, much the same as other MS-Windows applications have menus. In order to create menus, click on the menu icon on the toolbox or make the appropriate selection from the menu (Objects | Menu). The cursor will change its appearance to reflect the selection and will display as a crosshair with an attached menu icon. Move the mouse to the area in which you wish to drop the menu component. Press the left mouse button and drag open the component. A grid will appear and the first row will contain the entry ´Menu´. Placement is not critical on the page, as the component will not be visible at run time. On the other hand, it is critical on which page in the application it is placed. MindMap uses the menus it encounters, until it finds the next menu. Thus, it is important to consider their location, relative to the execution of your application. For example, this allows dynamic menus and context sensitive menus.

Menu Component: Attributes



For further information concerning the common attributes see Format Menu

Menu Component: Component Specific Attributes



Main menue [✓]
Sub menue
Popup Menue
Parse menue labels

Once you have placed a menu component, its default setting is **Main menu**. You do not need to edit this, even if you want it to be a sub menu. The setting will automatically change when two menu components are connected, using the rubber band approach.

The option **Parse menu labels** is used if you wish to have the contents of the menu entry be dynamic. Normally, you enter a static text which will display at run time. An alternative to this approach is to enter a component name and let the contents of the menu option become the contents of the referenced component. Thus, you can change the menu entry dynamically. In order for MindMap not to take the component name as a literal, you must specify that the menu entries are to be parsed. This is set in the **object specific** attributes.

This feature is especially handy, for example, if you are designing multilingual applications.

Creating a Menu

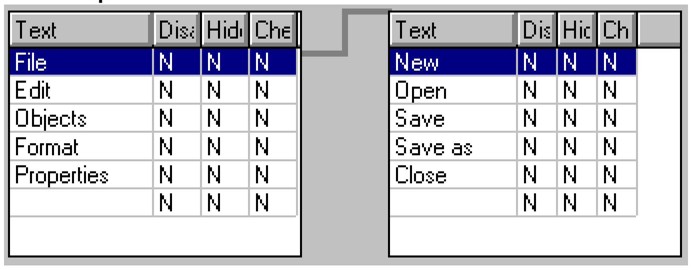
In order to create a menu, it is necessary to place a main menu component, as well as all corresponding sub-menu components. Every entry into a given menu component belongs to the same menu level.

Once you have placed the menu component, the first row in the displayed grid will contain the term ´Menu´. Select the row and begin to key in the new text..

Note:
You can easily create so called accelerator keys in the menu by prefixing the desired letter with a ´&´-symbol. This will result in the following letter to be displayed with an ´_´ symbol (underscore). At runtime, the menu option can be immediately executed from the keyboard by pressing the ALT - *key along with the appropriate letter. The menu option Exit would be executed by pressing* ALT+E.

Please terminate every entry with an *ENTER*. This will cause the creation of a new blank row. In case you do not intend to enter an additional menu option, you can just neglect this row; otherwise continue to enter the menu options.
All entries into one component will be displayed on the same menu level. If you desire parent-child menu structures, you must place a menu component for each of the siblings.

Example:



Text	Dis	Hid	Che
File	N	N	N
Edit	N	N	N
Objects	N	N	N
Format	N	N	N
Properties	N	N	N
	N	N	N

Text	Dis	Hic	Ch
New	N	N	N
Open	N	N	N
Save	N	N	N
Save as	N	N	N
Close	N	N	N
	N	N	N

In this example, two menu components have been placed on the desktop. The leftmost component is intended to represent the parent menu, whereas the right menu component is to become the child menu. This is represented by the connecting line between both components.

Note:
*Please note that the entry of "**.**" in a sub-menu will result in the displaying of a horizontal line (seperator). You can use this facility to group menu options into visible groups.*

After you have placed a main menu component and one or more sub-menu components, you can define the hierarchical relationship between these menu components by employing the rubber band technique. Place the cursor above the menu entry which you want to be the main level. Now press the left mouse button and, keeping it pressed, move over the appropriate sub-menu component. By doing so, you will see a rubber band attached to the cursor. Once you release the left mouse button, the two menu components are connected. This is represented with the connection line between the components. Proceed in this manner to define all other menu components.

If you wish to remove a rubber band connection between a menu and a sub menu, press the left mouse button on the option in the main menu and drag the mouse to a place on the screen other than above a sub menu. This will remove the previous connection.

If you wish to edit an existing menu option, select the appropriate entry by clicking the left mouse button on it. Next press *CRTL+E* (*edit*). Now you can proceed to edit the entry. Once the modifications have been completed, press the *ENTER* key.

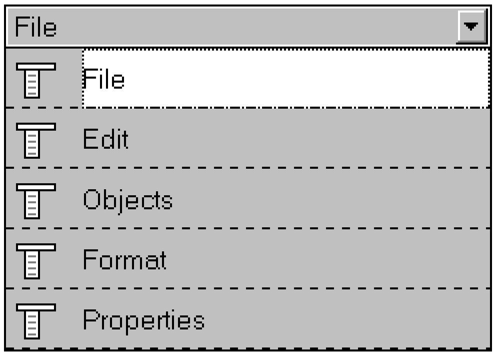
Note:
The rubber band automatically makes the sub-menu appear when the parent menu option is selected by the user at run time. You do not have to create a link for this procedure.

In the menu component, you can define whether your menu label should be **Disabled**, **Hidden** or **Checked**. **Disabled** implies that the menu option is grayed and thus not accessible. If a menu option is set to be **Hidden**, then it does not even appear in the menu as an option. If a menu option is set to be **Checked**, then a little check mark will appear, once it has been selected. If it is selected a second time (*or the user presses the* ESCAPE-*Key*), then the check mark is removed. This is a commonly used feature in MS-Windows applications and can be viewed in the MindMap **Edit | Set Tab order…** option.

Note:
*The options **Disable**, **Hidden** and **Checked** are changed at run time via the “Change attribute“-Message. (see section Links/ Messages/ Change attributes)*

Menu Component: Events

The menu component has only one type of event. These are the menu entries themselves. In the above example, these would be as follows:



Menu Component: Special Note

As the application executes, MindMap menus take effect beginning on the page on which they have been placed. Once a page, which contains different menu components, becomes visible, then these menus become valid, until another page containing menu components is encountered, and so forth. If you choose to place menu components on separate pages, you must assure that the page receives focus. This can be accomplished by jumping first to the page with the menus and then immediately jumping to the page displaying the actual user interface.

Note:
*When using menus, you should not run a MindMap application in full screen mode. (File | Preferences | Application**:** Run the application in full screen mode)*

Output Page Component

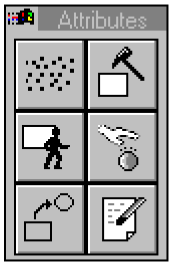
This component is used to print information. It presents itself as a blank sheet of paper. You can place a MindMap component on the output page in the same manner you would place the component on a MindMap page. The appearance of output pages can be influenced by the user at run time, if intended by the developer.

Since this component is not represented on the toolbox with an icon, you can only access it via the menu **Objects | Output Page**. Once you have selected this component, the appearance of the cursor will change to a crosshair with an attached page. In order to place the component, move the cursor to the screen position where you wish to place the top left corner of the page. Press the left mouse button and drag the mouse to the size for the page. Then let go of the left mouse button. A white rectangle will become visible. You will be prompted for a name for the page.

Note:
The output page component always retains the page metric or size ratio. Thus, you cannot create a page with an invalid height / width relationship.

As the name implies, this component is used to interact with the printer. Therefore, it is important to make the appropriate selection for the paper size (*US legal, DIN, etc*.). In case you change the size of the frame, the white rectangle inside the frame will still reflect the page metric. If you need to change the position of the component on the screen, grab it by clicking on the frame itself - not inside on the white rectangle.

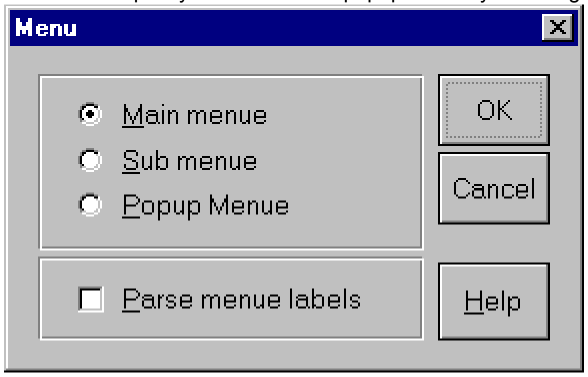
Output Page Component: Attributes



For additional information regarding common attributes Format Menu

Creating a popup menu

You can specify a menu to be a popup menu by selecting the appropriate radio button in the object specific dialog box.



Menu

[✓ Main menue]
Sub menue
Popup Menue

☐ Parse menue labels

OK
Cancel
Help

While normal (window oriented) menus are attached to the caption bar of the MindMap application they are defined in, popup menus show up anywhere on the screen, usually placed right besides the current mouse position.

Therefore a popup menu is not activated by displaying the page where it has been defined, but rather by issuing a jump link command to the menu component.

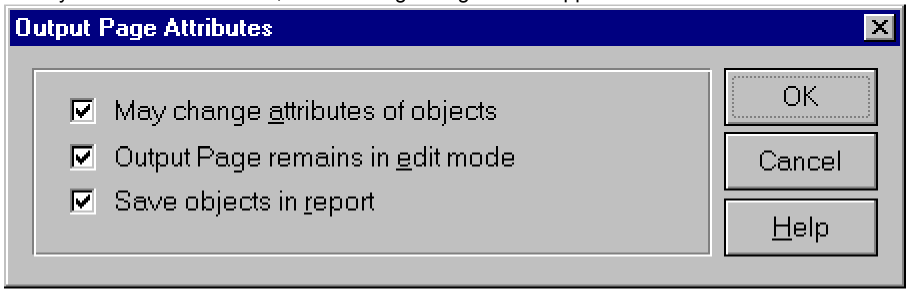
Left mouse button released: Jump to   mnu1

This will cause the specified menu to appear as a popup menu as close as possible to the current mouse position.

Note:
There is no need to have the menu on the same page that is currently active while this function is performed.

Output Page Component: Component Specific Attributes

If you select this attribute, the following dialog box will appear:



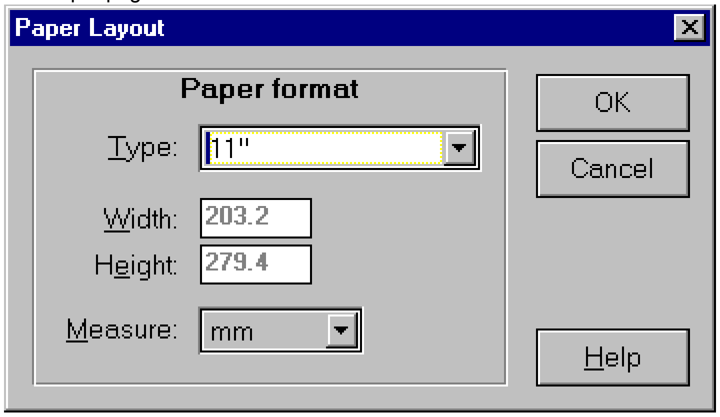
Output Page Attributes

☑ May change attributes of objects
☑ Output Page remains in edit mode
☑ Save objects in report

In the object specific attributes of output page components, you can check several options:
May change attributes of components: This allows the user to change the attributes of components placed on the output page at runtime. This option must be viewed in conjunction with the option **Output Page remains in edit mode**.
If you do not permit editing at run time, the setting of the aforementioned option is invalid.

Output Page remains in edit mode: This option allows the user to change aspects of the output page at run time. By enabling mouse-related drag & drop operations a user may add e.g. graphic components or input field components to the output page at run time.

Save components in Output Page: This option should be set when it is desired that components on the output page component are to be stored in the application itself. This should always be set in cases where the user will not be defining the output page at run time.



Paper Layout

Paper format

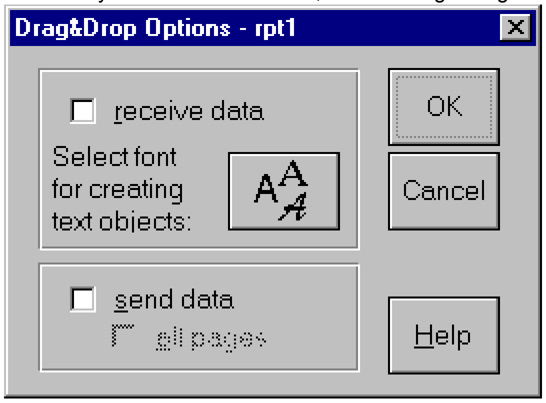
Type: 11"
Width: 203.2
Height: 279.4
Measure: mm

In the **Type** list, you can choose one of the American / English formats or one of the European DIN formats. Depending on your choice, the **Width** and **Height** numbers will be adjusted. By choosing **User defined,** you are able to key in your own width and height. In the **Measure** box, you can choose the appropriate unit of measurement.

Note:
The size of the output page that you have painted on the screen will not change; however, the actual printed output page will be adjusted to your choice. In this case, you will have to resize the output page component on the screen by hand.

Output Page Component: Drag & Drop

Once you select this attribute, the following dialog box will appear:



This dialog box contains an icon for the font attribute. Selecting this attribute will result in the displaying of the common font dialog box.

Note:
By means of the drag & drop option, you can have text components created at run time on the output page component.

An input field with the drag & drop option set to **send data is** required. The output page must have its drag & drop option set to **receive data**. At run time, the mouse is placed above the input field. Next press the left mouse button and move the mouse to the position over the output page to where the text is to be copied. Once the left mouse button is released, a text component is created and dropped on the output page. It is automatically set to contain a formula and its value is the text contained in the original input field.

Note:
Components created in this manner cannot be stored in the application. They only exist as long as the application is in run mode.

Output Page Component: Additional Events

The output page component does not register additional events.

How to use an Output page component for printing

You can place a variety of components on an output page, just as you can place them on any MindMap page.

Note:
In some cases it does not make sense to place certain components on the output page. For example, placing an input field on an output page does not make sense, since user interaction with it is not possible.
Only the following components can be placed on an output page: text, data table, graphic. The creation of all other components on the output page is prohibited.
If you wish to place text on the page, choose the text component. You can also connect components on the output page to external data sources.

Let's assume you want to place a text component and a bitmap on the output page. Pick up a text component from the toolbox and place it on the output page component. Note that you cannot key in text as you can when placing a Text component on a regular MindMap page.

Note:
Text components that are placed on an Output page automatically become formula fields.
Please open up the formula editor or assign a value to the appropriate text component using a link message (Text Component/Component Specific Attributes)

Σ

For this purpose, you have to use the formula function. You can find it by clicking on the Sum symbol in the status bar at the bottom of the screen. Select the text component, click on the Sum symbol and key into the formula field the text you want to see displayed by the Text component. Don't forget to enclose the text in quotation marks, for example "text". (Parser)

Pick up the graphic import component from the toolbox**.** Use it to place any bitmap on the output page component. (*You'll find several bitmaps in the MS-Windows directory of your system, for example*)

Now let's place two other components on the MindMap page, right beneath the output page component. First, place one command button on the MindMap page (Button Component) . Next, click on the input/output button of the toolbox and click on the printer symbol of the input/output dialog box. Don't worry about the other options in this message box; just click the OK button to accept the settings and close the box. Now, place the printer component on the MindMap page (*not on the output page component*) and key in a name when you are asked for it by MindMap (Input/Output Component) .

Now, let's add the printing functionality: Select the command button, activate the attribute toolbox and click on the links button. Create a new link with the event "left mouse button released" and the message "Drag & Drop". In the "Move Drag & drop Data from" section, click on the symbol of your output page component. In the "Move Drag & Drop Data to" section, click on the symbol of your printer component. Close the dialog box by clicking on the OK button in the upper right corner. The resulting link should be:

Left mouse button released: Drag & Drop rep1 -> prt1

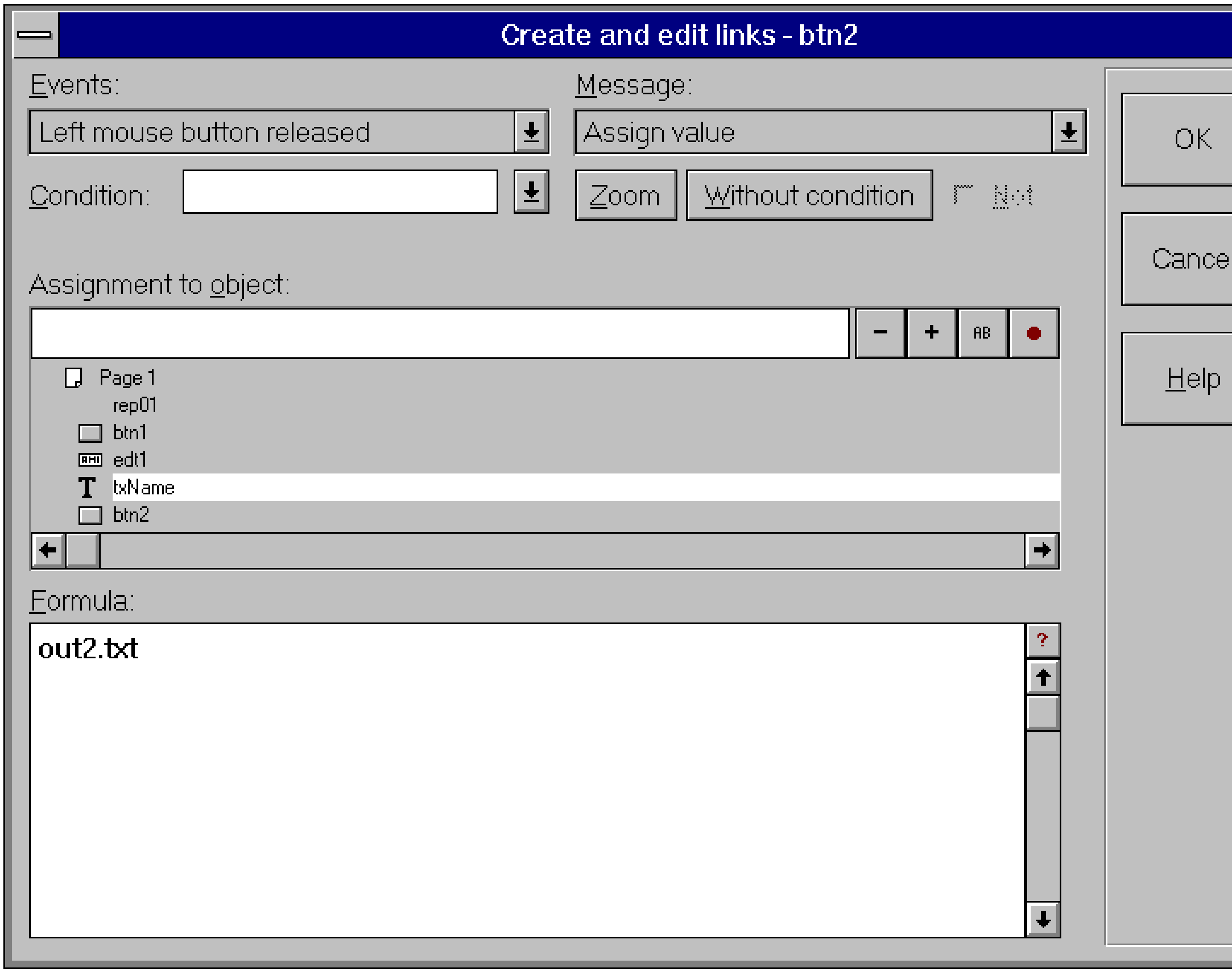
Now, change to the run mode of MindMap and click on the command button. If everything has been done as described, your printer will print out a page with the text of your text component and the bitmap.

Note:
Depending on how the application has been designed, the user might actually never see the output page. Usually, the print functionality is executed by simply pressing a button. The application never visibly jumps to the page containing the output page.

Output Page Component: Special Remark

Text components and data table components, that have been placed on an output page, will print on subsequent sheets of paper if there is not enough space to print their contents on the first page. Data tables automatically wrap to the next page. Text components must have their **Auto print multiple pages** attribute set. (Text Component/Component Specific Attributes) .

It is possible to use components in formulas, which have been placed on output pages (Parser). The notations **<report name>.<component name>** or
<report name>:<component name> are references to a particular named component on an output page. Following is an example of how the content of a text component on an output page is referenced by a text component on a MindMap page:



out2.txt

MCI Component

Based on the MCI (*Media Control Interface*) specifications, MindMap offers a multimedia component that serves as a vehicle to display and maintain Video For Windows (*AVI*) files, Sound (WAV), as well as other types of multimedia devices in MindMap applications.

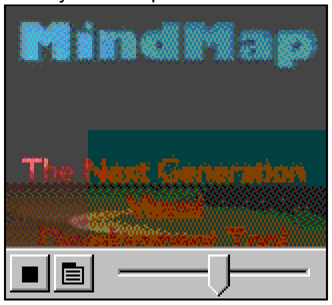
To create a multimedia Component, select the appropriate icon from the toolbox or in the menu **Objects | MCI Object** and draw the component at its desired position and size. Since all multimedia devices share a common interface, this interface is implemented as MindMap link messages, as far as commands to the multimedia component are concerned. An object specific attributes dialog to alter the appearance of the component is available, as well. This feature allows you to open all types of MCI components. In the select dialog box, all the file extensions are listed that can be loaded. They are as follows: "*.wav; *.mid; *.rmi; *.avi; *.mmm; *.mov; *.pic; *.jpg; *.flc; *.fli; *.aas and *.mp2". In the dialog box, only the files with these extensions will be listed.

Note:
The MindMap MCI component relies on the existence and functionality of special MCI drivers. Make sure that these drivers are properly installed before you create an MCI component.

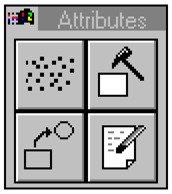
If you are running MindMap on MS-Windows 3.1, 3.11 or MS-Windows for Workgroups you must install Video for Windows (*VfW*). Refer to the corresponding documentation supplied with Video for Windows to learn how to install it. If you are running MindMap on MS-Windows 95 VfW is part of the operating system. Make sure that it has been installed through the **Add/Remove Software** applet in the control panel.

If you choose to install additional multimedia software which interfaces with MCI, you will possibly be able to interact with more file types.

If you have placed an MCI component, e.g. an AVI file, it will display itself as follows:

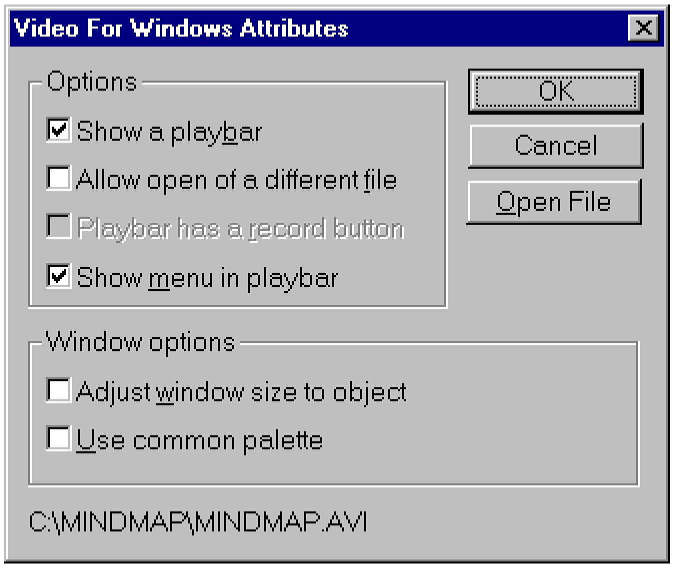


MCI Component: Attributes



For additional information concerning the common attributes: Format Menu

MCI Component: Component specific Attributes


Video For Windows Attributes

Options
- [✓] Show a playbar
- [ ] Allow open of a different file
- [ ] Playbar has a record button
- [✓] Show menu in playbar

Window options
- [ ] Adjust window size to object
- [ ] Use common palette

C:\MINDMAP\MINDMAP.AVI

The following options are defined through the object specific dialog:

Option	Description
Show a play bar	Defines that the component should have a play bar and at least a start / stop button.
Allow open of a different file	Defines that the user is allowed to open a new multimedia component through an **open** command in the menu (*if it exists*).
Show menu in play bar	Defines that a pop-up menu should appear if the right mouse button is pressed over the surface of the component or, if a play bar exists, a menu button should be visible.
Play bar has a record button	Defines that a record button should appear for devices that are capable of recording information (e.g. wave audio).
Adjust window size to object	Defines that the size of the Multimedia component is controlled by the size of the MindMap component instead of being auto-sized depending on the size of the video source.
Use common palette	If this option is checked, the common palette defined by MindMap is selected for the Video For MS-Windows component, instead of letting VFW define it's own palette. Setting the graphic component to use a common palette as well, is useful if a video has to be played concurrently with a graphic on the same screen. Note that this applies only to 256 color palletized graphic modes. Also note that specifying a common palette may result in some loss of visual quality.

MCI Component: Additional Events

The links dialog box for multimedia components has some of the standard events. The additional events are shown in the following table:

Event	Description
Mode changed	This event is generated if the mode of the multimedia component changes (*e.g. playing, seeking, stopped, ...).*
Device not ready	This event is generated if the device is in a state where playing a multimedia file is not possible, as is the case if a CD was ejected from a CD drive.
Stopped	This event is generated when the device ends playing a multimedia sequence, regardless of whether the device stops through user interaction or when the end of the media is reached. Prior to sending this event, the **Mode changed** event has been sent.
Playing	This event is generated when the device starts playing a multimedia sequence. The event is sent after the **Mode changed** event has been fired.
Recording	This event is generated when the device starts recording a multimedia sequence. The event is sent after the **Mode changed** event has been fired.
Seeking	This event is generated when the device moves to a new position of a multimedia sequence. The event is sent after the **Mode changed** event has been fired.
Paused	This event is generated when the device is in pause mode. The event is sent after the **Mode changed** event has been fired.
Device is open	This event is generated when a multimedia sequence is ready to be launched after loading. Prior to sending this event, the **Mode changed** event has been sent.
Position changed	This event is generated continuously once every second to indicate the progress of playing a multimedia file. Use the mciGetPosition or mciGetPositionString functions to retrieve the current position. (Parser)

Size changed	This event is generated when the MCI interface changes the size of the display window.
Media changed	This event is generated when a new media (*such as a CD*) is loaded.
Error	This event is generated when the MCI interface detects an error. The formula value of the MCI component contains the appropriate error string.

Visual Basic Component

This component allows the integration of Visual Basic Custom Controls. These so called VBX Controls are embedded in MindMap, similar to other MindMap components, with a common user interface and messaging protocol. Depending on the capabilities of a particular VBX control, only the command set (*here called properties*) varies from control to control. Since MindMap does not distinguish between the Visual Basic type of run and edit mode, from the control's point of view, the system appears to be in run mode all the time. Please note that controls that behave differently when in VBX edit mode, may not work properly in MindMap.

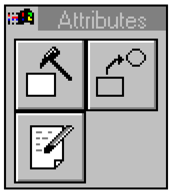
In so far as VBXs have been installed, (File | Preferences | VBX), you will see their graphical representation on the toolbox, given that they have registered themselves with an icon. You will also see them entered on the menu **Preferences | VBX**. Once you have selected a VBX component, the cursor changes to a crosshair with the letters VBX attached. Move the mouse to the screen position where you wish the VBX to appear. Press the left mouse button and drag the mouse until the dashed rectangle has the desired size. Release the mouse button.

Installing a VBX Custom Control

VBX files are attached to MindMap by selecting them in the preferences dialog box (File | Preferences | VBX). Once installed, you must shut down MindMap and restart it, in order to load the new control into the menu and toolbox and to make it accessible (File | Preferences | VBX). Every installed VBX has at least an entry in the menu. Generally, VBXs also register an icon on the toolbox.

Note:
MindMap emulates portions of Visual Basic. To be installed into the MindMap environment, it is required for a VBX to comply to the Visual Basic API Version 1.0. In general, if a VBX uses functionality of Visual Basic Versions 2.0 or higher, this particular VBX cannot be installed in MindMap. In this case it either displays a dialog box or it may behave unpredictable. Since the environment appears to be in run mode from the VBXs perspective, the VBX must support dynamic creation in run mode. Some VBXs require to be inherited from a Visual Basic Form when Visual Basic switches to run mode. These VBXs might not behave properly in MindMap.
Also, since MindMap's drag & drop metaphor is completely different from Visual Basic's, drag & drop functionality supplied by a VBX is not supported in MindMap.

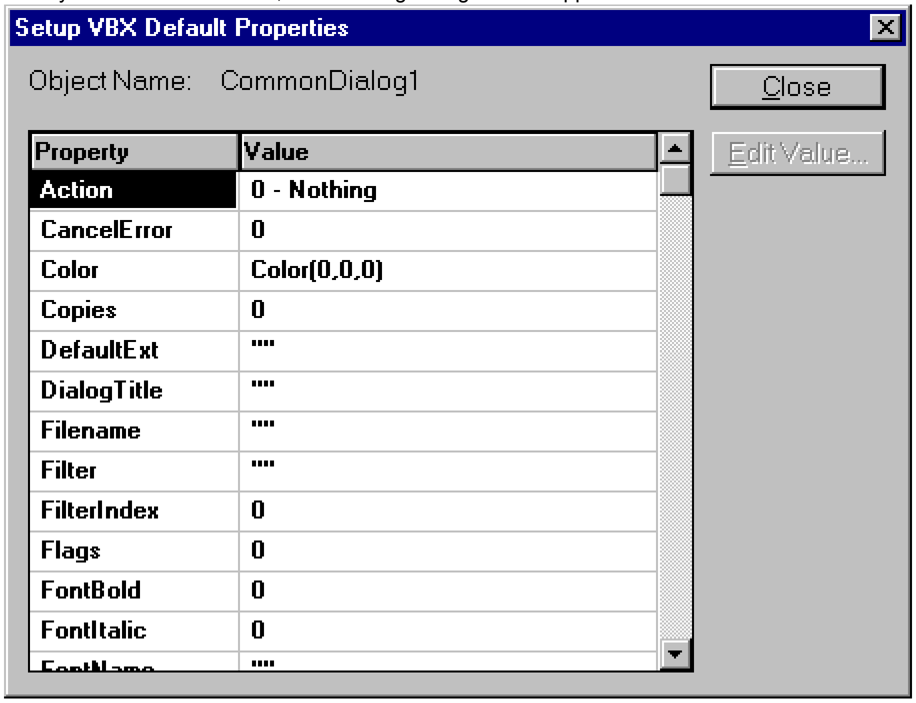
Visual Basic Component: Attributes



For further information regarding common attributes
Format Menu

Visual Basic Component: Component Specific Attribute

If you select this attribute, the following dialog box will appear. The actual entries in the list vary depending on the VBX.



Object Name: CommonDialog1

Property	Value
Action	0 - Nothing
CancelError	0
Color	Color(0,0,0)
Copies	0
DefaultExt	""
DialogTitle	""
Filename	""
Filter	""
FilterIndex	0
Flags	0
FontBold	0
FontItalic	0
FontName	""

All properties that have the PF_RunmodeW and PF_RunmodeR bits set, can be edited through the VBX's object specific attributes dialog. This dialog is common to all VBX controls. The left column lists the **Property** name in alphabetical order and the right column contains the actual **Values**. These values may be set by double-clicking the left column *(i.e. the property name*) or by simply typing the desired value. Some properties allow you to make the value selection from a predefined list. You can recognize this if the **Edit Value…** button is activated when you select a property. Please note that all property values are parsed. This means that parser statements containing the names of other components or MindMap parser functions may be used. Be sure to surround strings with double quotes.

Once a property is edited, the VBX control should reflect the change, even if this dialog is not closed.

Also note that all properties set through this dialog are saved into the application file and automatically restored at load time.

Of course, you may alter the properties of a VBX component at run time by creating a link and specifying what property should receive which value. Similar to most other MindMap components, VBX components also register a set of events which appear in the list of events in the link dialog, at the bottom of the list. Please note that all events that a VBX might execute are listed, but that some events may appear in this set which do not work in the MindMap environment (*e.g. dragging events*).

Notes:
The VBX controls can't be described in more detail because there are thousands of controls available. For more details on the controls, you should consult the manual of the particular VBX that you wish to use. Also note that MindMap can only change or modify those aspects of the VBX's properties that the original VBX developer has allowed us to access.

Properties

Communication with VBX Controls is accomplished by getting and setting properties. Property types are either:

Strings

Strings are null-terminated ASCII strings compatible with the concept of MindMap. They are limited in size to 16383 bytes.

Numericals
(*multiple types, but mapped to the single MindMap number type*)

All types supported by VBX controls are converted to and from the only number type used in MindMap.

Enumerations

Enumerations are numbers as well. For the user’s convenience, their value may be selected from a list in some of the dialogs.

Pictures

Pictures are currently not supported.

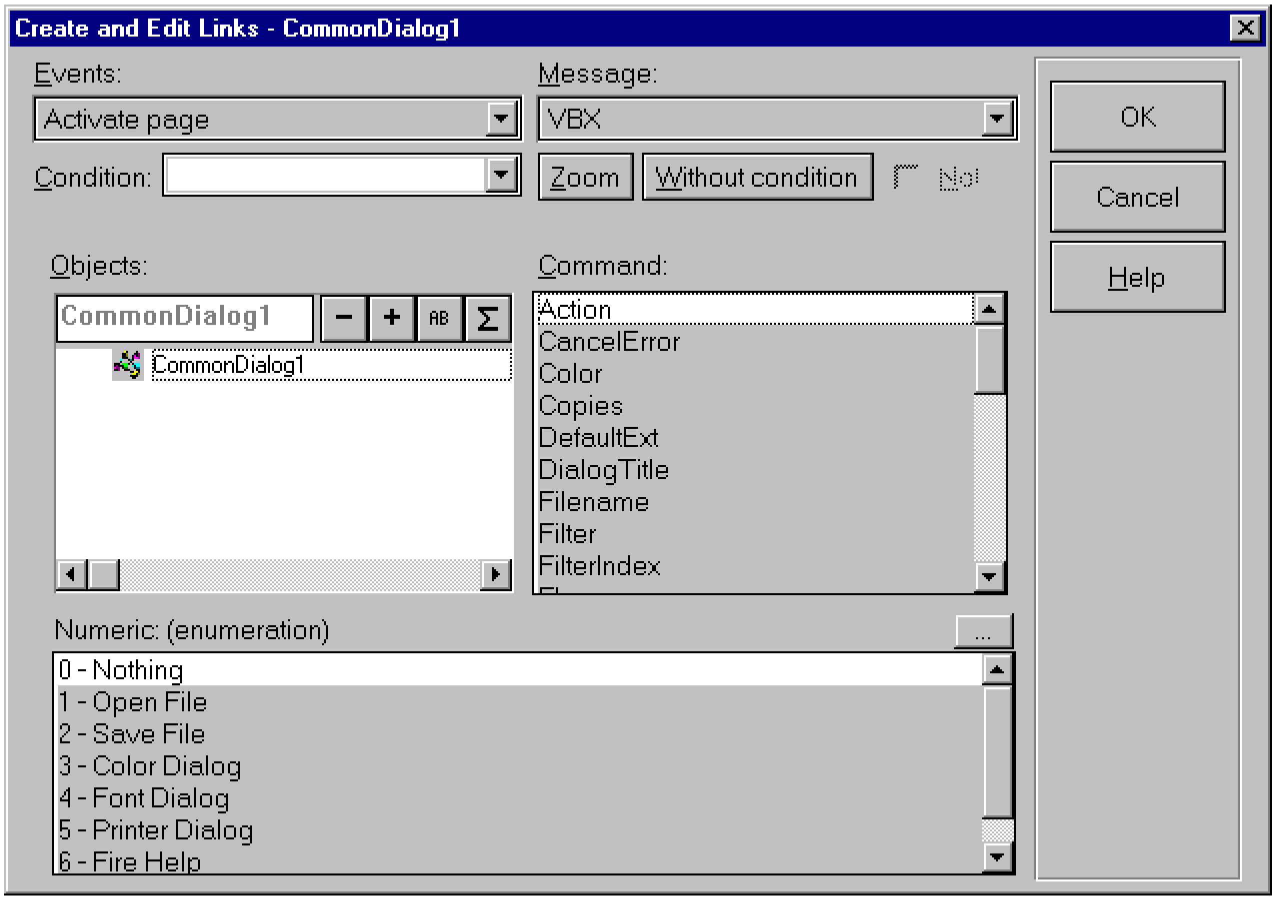
Visual Basic Component: Additional Events

Similar to most other MindMap components, VBX components also register a set of events which appear in the list of events in the link dialog. Please note that all events that a VBX might execute are enumerated, so that some events may appear in this set which do not make sense in the MindMap environment (*e.g. dragging events*).

Some VBX implementations may pass arguments via   events. Please use the *GetParam* function registered by the VBX component to access these parameters. Refer to the Parser documentation.

Communication

Of course, you may alter the properties of a VBX component at run time by creating a link and specifying what property should receive which value.



Create and Edit Links - CommonDialog1

Events: Activate page

Message: VBX

Condition:

Objects: CommonDialog1

Command:
Action
CancelError
Color
Copies
DefaultExt
DialogTitle
Filename
Filter
FilterIndex

Numeric: (enumeration)
0 - Nothing
1 - Open File
2 - Save File
3 - Color Dialog
4 - Font Dialog
5 - Printer Dialog
6 - Fire Help

The figure displays a link on the command button. If the button is clicked, attributes will be changed.

Note:
The link dialog box for the messages relating to the VBX contains properties, which are displayed as commands. You enter the values in the lower portion of the dialog box.