File Menu

New...

Starts a new SmartObject file. Allows you to start the file in one of three modes: Page Layout, Formatted Text, or Text (ASCII).

Related Topics:

Starting New SmartObject Files
Using Formatted Text Mode
Using ASCII Mode

Open...

Opens an existing SmartObject file.

Related Topic:

Opening Existing SmartObject Files

Save

Saves an existing SmartObject file with the same name.

Related Topic:

Saving Existing SmartObject Files

Save As...

Allows you to save and name a SmartObject file for the first time, or to save and rename an existing SmartObject file.

Related Topic:

Saving New SmartObject Files

Delete

Allows you to permanently delete the current SmartObject file.

Page Setup...

Sets the top, bottom, left, and right margins and specifies information to be included in the header and footer of a printed page.

Print...

Prints one or more pages in the current SmartObject file.

Related Topic:

Printing SmartObject Files

Printer Setup...

Allows you to select a printer driver and a printer connection.

Set Path From File...

Causes the SmartObject Editor to read the .PTH file of an IconAuthor application. The .PTH file tells the SmartObject Editor where the files for that application are located.

Related Topic:

Path Information

Exit

Exits the SmartObject Editor.   Lets you save unsaved changes before quitting.

Related Topic:

Quitting the SmartObject Editor

Edit Menu

Cut

Removes the selected object or text.   A copy of the removed item is placed on the Clipboard.

Related Topics:

Cutting Objects
Cutting Text

Copy

Copies the selected object or text onto the Clipboard.

Related Topics:

Duplicating Objects
Copying Text

Paste

Inserts the contents of the Clipboard onto the current page.

Related Topics:

Duplicating Objects
Pasting Text

Clear

Clears the selected text or object.   Cleared items are permanently removed and are *not* placed on the Clipboard.

Related Topics:

Clearing Objects
Clearing Text

Copy To...

Copies the selected text in a text object to a specified ASCII text file.   If that file existed previously, its previous contents are overwritten.

Paste From...

Inserts the contents of an ASCII text file into a text object.

Related Topic:

Creating Text Files

Page Menu

New Page...

Allows you to create a new page in the current SmartObject file.Allows you to create a new page in the current SmartObject file.

Page Maintenance...

Displays a the Page Maintenance dialog box which lets you: create a new page, rename a page, copy or delete a page, or import a page from another SmartObject file.

Related Topics:

Creating New Pages
Copying Pages
Deleting Pages
Importing Pages
Changing Pages
Renaming Pages

Page Properties

Displays a cascading menu of commands that let you set certain characteristics of the current page.

Color...

Allows you to set the color of the current page.

Related Topic:

Working with Page Color

Page Name...

Allows you to name the current page.

Next

Displays the next page in the SmartObject file.

Previous

Displays the previous page in the SmartObject file.

Go To...

Allows you to jump to a particular page in the file.

Objects Menu

Tools

Displays a cascading menu of tools that let you select or create different kinds of objects.

Select Object

Causes the Select Object tool to be in effect. Depending on whether you select one object or a group of objects, different editing tasks become available. For example, you can cut, copy, paste or resize one selected object. You can move or align multiple objects or set a common property such as FamilyName.

Related Topic:

Selecting and Editing Objects

Button

Allows you to create button objects such as push buttons, check boxes, and radio buttons.

Related Topic:

Button Objects

Graphic

Allows you to create graphic objects. Once you draw a graphic object you can set its properties, for example, so that a particular graphic is displayed.

Related Topic:

Graphic Objects

Listbox

Allows you to create listbox objects. Once you draw a listbox object you can set its properties, for example, so that a particular list of items is displayed.

Related Topic:

Listbox Objects

OLE

Allows you to create OLE objects. OLE objects take advantage of the Microsoft Windows OLE (Object Linking and Embedding) feature. From within the SmartObject Editor you can access any Microsoft Windows application that is OLE-ready and create data that eventually becomes an object within your IconAuthor application.

Related Topic:

OLE Objects

Text

Allows you to create text objects. Once you draw a text object you can set its properties, for example, so that it is editable. If a text object is editable, a user can type in it at runtime. If a text object is not editable, the author types text in it within the SmartObject Editor and the information is display-only at runtime.

Related Topic:

Text Objects

Timer

Allows you to create timer objects. Once you draw a timer you can set its properties, for example, so that it is set to count down, count up, go off periodically, or go off at a particular time of day.

Related Topic:

Timer Objects

Transparent

Allows you to create transparent objects. A transparent object acts like a push button that cannot be seen. The transparent object can cover a small portion of a page or an entire page.

Related Topic:

Transparent Objects

Object Properties

Displays a cascading menu of commands that let you set the properties for the currently selected object. The items in this menu vary depending on the object that you select. (The items in this menu are identical to the items that appear in the pop-up menu when you right mouse click on an object.)

Properties...

Allows you to display the Properties dialog box for the selected object. The list of properties varies depending on the object that you select.

Related Topic:

Setting Object Properties

Object Type...

This menu item is only available for text objects, graphic objects, and OLE objects. Allows you to indicate whether an object is live or static. When an object is live its properties can change at runtime. Otherwise, it behaves as if it is an unchanging part of the background of the display. (This item only appears for a limited number of objects because all other objects are *always* live and cannot be made static.)

Related Topic:

Making an Object Live or Static

Button Styles...

This menu item is only available for button objects. Allows you to specify the style of a button: push button, icon button, check box, radio button, or group box. By default, the property list for a button object is for push button style. If you change the style of the button, the property list also changes.

Item List...

This menu item is only available for listbox objects. Allows you to specify the items that you want to appear in a listbox object.

Related Topic:

Entering Items into Listboxes

Insert New Object...

This menu item is only available for OLE objects. Allows you to insert data (from an OLE server) into the currently selected OLE object.

Related Topic:

OLE Objects

Convert Data to Static...

This menu item is only available for OLE objects. Allows you to convert the *data* in an OLE object to static. When the data is static it means that it cannot be changed because the connection to the server has been permanently severed. Static data is different from a static object (an object that is not live). An OLE object with static data can be live and still have its properties, such as visible and enabled, changed at runtime.

Paste

This menu item is only available for OLE objects. Allows you to paste data (such as a picture or selected cells from a spreadsheet) that was previously cut or copied to the Clipboard (from a server application) into your OLE object.

Paste Links

This menu item is only available for OLE objects. Allows you to paste server data from the Clipboard into an OLE object. The data is linked to the file from which it was originally copied. As a result, any time that you open the server application and change and save that file, the changes you make are automatically updated in the OLE object on the SmartObject page.

Links...

This menu item is only available for OLE objects with which you have previously used the Paste Link... command. The Links... command displays the Links dialog box that describes the current status of the OLE link, such as the server name and filename.

Block Styles...

This menu item is only available when a text object is selected. Allows you to specify the certain characteristics of the text object such as colors, shading, and shadow.

Related Topic:

Working with the Style of a Text Object

Input Styles...

This menu item is only available when a text object is selected. Allows you to specify the kind of characters that can be entered in a text object (by the author and/or the user) and in some circumstances, the positioning and quantity of text allowable.

Timer Styles...

This menu item is only available for timer objects. Allows you to specify the style of a timer: Periodic, Count Down, Count Up, or Alarm.

Fonts...

This menu item is only available when you are editing text in a text object. Allows you to change the font information (such as font type, size and color) for text.

Text Color...

This menu item is only available when you are editing text in a text object. Allows you to change the color for text.

Fill Color...

This menu item is only available when you are editing text in a text object. Allows you to change the color used for the background of text.

Text Styles...

This menu item is only available when you are editing text in a text object. Allows you to work with the named styles used to set the characteristics of text. You can select, edit, or remove an existing style or you can create a new style.

Left

This menu item is only available when you are editing text in a text object. Allows you to left align the text in the current paragraph.

Center

This menu item is only available when you are editing text in a text object. Allows you to center the text in the current paragraph.

Right

This menu item is only available when you are editing text in a text object. Allows you to right align the text in the current paragraph.

Single

This menu item is only available when you are editing text in a text object. Allows you to single space the text in the current paragraph.

1 1/2

This menu item is only available when you are editing text in a text object. Allows you to one and one-half space the text in the current paragraph.

Double

This menu item is only available when you are editing text in a text object. Allows you to double space the text in the current paragraph.

Alignment

This menu item is only available when a group of objects is selected. Displays a cascading menu of commands that let you change the alignment of the objects in the group.

Left

All objects are moved so that their left side is aligned with the left-most selected object.

Right

All objects are moved so that their right side is aligned with the right-most selected object.

Top

All objects are moved so that their top border is aligned with the top-most selected object.

Bottom

All objects are moved so that their bottom border is aligned with the bottom-most selected object.

Center Vertical

All objects are moved so that they are centered on a common vertical axis.

Center Horizontal

All objects are moved so that they are centered on a common horizontal axis.

Order

This menu item is only available when an object is selected that is at least partially obscuring one or more other objects. Displays a cascading menu of commands that let you change the position of the object in the stack.

Bring Forward

In a stack of objects, brings the currently selected object one position closer to the front.

Send Backward

In a stack of objects, brings the currently selected object one position away from the front.

Bring To Front

In a stack of objects, brings the currently selected object to the front.

Send To Back

In a stack of objects, brings the currently selected object to the back.

Options Menu

Full Screen

When this option is toggled on, the SmartObject Editor appears full screen. (The menus, ribbon bar and status bar appear in a floating window that can be moved around the screen.)

Hide Tools

This option is available when the SmartObject Editor appears full screen. It allows you to hides the menu bar (and the ribbon and status bars). To make the menu bar visible again press CTRL + D.

Grid Alignment...

Displays the Grid dialog box. The Grid dialog box lets you toggle the grid feature on and off, make the grid visible or not visible, and change the cell dimensions used for the grid.

Related Topic:

The Grid

Confirm Clear

When this option is toggled on, it causes the SmartObject Editor to ask you to confirm whether you want to clear an object when you use the Clear command from the Edit menu.

Overlay Mode

When this option is toggled on the current video
frame is displayed on the screen. (This feature is only available if the appropriate video hardware has been installed and set up.)

Related Topic:

Displaying Video in the SmartObject Editor

Ribbon Bar

When this option is toggled on the ribbon bar is visible.

Status Bar

When this option is toggled on the status bar is
visible.

Save Settings On Exit

When this option is toggled on and you exit the SmartObject Editor, settings that describe the condition of the editor (such as the size of the window or the grid settings) are remembered the next time you restart the editor.

Help Menu

Index

Displays an alphabetical list of all Help topics that are available.

Keyboard

Displays the accelerators for performing some tasks with the keyboard instead of the mouse.

Commands

Displays an explanation of commands.

Procedures

Displays a description of how to use IconAuthor

Using Help

Displays a short tutorial and other information about using Windows online Help.

About SmartObject...

Displays SmartObject Editor copyright and version information.

Objects

Each kind of object is called a class.   The following object classes are available in the SmartObject Editor:

button
graphic
listbox
OLE
text
timer
transparent

You include these objects in your file by choosing the appropriate object tool, drawing the object on the page, and then defining the object.

All objects have certain properties (characteristics) that you define in order to make them appear and perform in a particular manner.   For example, a button object has a property called Label that allows you to change the text label that appears on the button.   Buttons also have a property called Style that lets you specify what kind of button you want, for example, a push button, a radio button, or a check box.

The most important property of an object is whether it is static or live.   Many objects can only be live, while text, graphic, and OLE objects can be live or static.

Related Topic:

Making an Object Live or Static
Creating Objects
Setting Object Properties

Live Object Maintenance

If you plan to create live objects in your SmartObject files it is important that you understand how to take care of those objects at runtime. For example, once a live object exists, you can use an ObjSet icon in IconAuthor to reset its properties.

Another important consequence of using live objects is that once you create them on the SmartObject page, and display the page in IconAuthor, the objects continue to exist in memory until you deliberately remove them. Once there is no longer a need for an object, you must use an IconAuthor ObjDelete icon to delete it permanently. When you use an ObjDelete icon you remove the object from memory but you don't effect the original object in the SmartObject file. In fact, once an object is deleted at runtime, the only way to recreate it is to display the original SmartObject page again.

Creating Objects

Use the object tools to draw objects. By default, most of the objects that you draw are initially white-filled with a thin black border. The exceptions are the timer object that appears as a clock, the button object that appears as a push button, and the transparent object which has horizontal lines across it. By default, when you draw objects they are conforming to an invisible grid in the work area.

When you begin drawing keep in mind that you don't have to draw the object in the precise dimensions or position because you can always select an object and then resize, move, or even delete it. The maximum size of objects is defined by the dimensions of your work area. You should also be aware that drawing objects is typically just the first step in making the objects appear and perform as you want. For example, once you draw a text object you still need to enter text in it and once you draw a button you still need to change its text label so that it says OK, Cancel, etc. to meet the needs of your application.

You can create objects that are independent of the other objects on the page or you can create objects that overlap or completely hide other objects.

To draw an object:

1. Click on the Object Tools button in the function ribbon.

   A menu of the available object classes appears.

2. Click on the tool that represents the object you want to create.

   Note that the first menu option, Select Object, is not a tool for creating objects, but rather for selecting objects. When you select an object tool the cursor changes to a cross-hair.

3. Position the cursor in the work area at the point where you want the upper left corner of the object to appear.

4. Press and hold the left mouse button and drag diagonally down to the right until the cursor reaches the location for the lower right corner of the object.

   Note: If you begin drawing an object and decide not to continue there is an easy way to stop the process. While continuing to press the left mouse button click the right mouse button. The process is canceled.

5. Release the mouse button.

   The object is drawn.

You can continue to create new objects of the chosen class. If you want to draw another class of object, choose a different tool from the menu.

Object Alignment

In many situations, it is important that objects are in precise alignment with one another. For example, you may want two graphics centered on the page or four buttons aligned along the right side of the page. The SmartObject Editor comes with two features to aid in aligning objects: the grid and the Align Objects button.

The Grid

By default, when you draw objects they are conforming to an invisible grid in the work area. The grid makes it easy for you to align objects such as a vertical column of on-screen buttons. You can make the grid visible or invisible. You can also change the dimensions of the grid cells or simply disable the grid altogether. Use the Grid dialog box to change the grid options.

To change grid options:

1. Click on the Set Grid button.

   The Grid dialog box appears.

2. As necessary, make changes to the grid options.

   Grid Size

   By default, the grid lines define cells that are 10 x 10 pixels. Type new numbers in these text boxes to redefine the size of the cells.

   Snap To Grid

   By default, the grid is active. To disable the grid click on the Snap To Grid option to toggle it off.

   Show Grid

   By default, the grid is active but invisible. To make the grid visible click on the Show Grid option to toggle it on. When this option is on, points appear on the screen to identify the location of the lines that define the grid.

3. Choose OK to close the Grid dialog box.

The Align Objects Button

Regardless of whether the grid is in effect, you have the option of using the Align Objects button to align objects. In effect, you draw a box around a group of objects to select them and then click on the Align Objects button to show a menu of alignment options. You choose an option, such as left (to align all selected objects with the left-most selected object) or right (to align all selected objects with the right-most selected object).

To align objects with the Align Objects button:

1. Make sure that the Select Object tool (the arrow cursor) is active by clicking on the background.
2. Press and hold the left mouse button above and to the left of the objects you want to align.
3. Drag down and to the right to draw a box that encloses the objects.
4. Release the left mouse button.

   When you release the left mouse button the box disappears and the objects within the box are selected. (Each object is surrounded by eight small sizer blocks.)

   And, the Align Objects button becomes active in the function ribbon.
5. Click on the Align Objects button.

   A menu of alignment options appears.
6. Choose the desired alignment option and the selected objects are realigned.

The following paragraphs describe the effect created by the different alignment options.

<u>**This option:**</u>	**Moves all selected objects so that:**
Left	their left side is aligned with the left-most selected object
Right	their right side is aligned with the right-most selected object
Top	their top border is aligned with the top-most selected object
Bottom	their bottom border is aligned with the bottom-most selected object.
Center Vertical	they are centered on a common vertical axis
Center Horizontal	they are centered on a common horizontal axis

Selecting and Editing Objects

You can select one object or several objects. Once you select an object you can edit it. However, different editing tasks are available for single and multiple objects.

To select a single object:

Click on the object.

A selected object is immediately surrounded by eight sizer blocks; one at each corner and one in the center of each side. Event if the object you select is partially obscured by another object the eight sizer blocks appear, defining the selected object even though it is still obscured.

There is only one situation where you cannot just click on an object to select it. If you are editing text in a text object and decide to select that text object you can't just click on it. You have to click on some other object (or the page background) and then click on the text object you were editing.)

To select multiple objects:

1. Make sure that the Select Object tool (the arrow cursor) is active by clicking on the background.
2. Press and hold the left mouse button above and to the left of the objects you want to select.

3. Drag down and to the right to draw a box that encloses the objects.
4. Release the left mouse button.

   When you release the left mouse button the box disappears and the objects within the box are selected. (Each object is surrounded by eight small sizer blocks.)

Related Topics:

Resizing Objects
Moving Objects
Duplicating Objects
Removing Objects
Aligning Objects

Resizing Objects

After you draw an object you usually have the ability to resize it. (The graphic and OLE objects are the exception.) Frequently, it's practical to create an object, set some of its properties, and then resize it as necessary.

To resize an object:

1. Select the object.
2. Position the cursor over one of the eight sizer blocks.
3. Press and hold the left mouse button and drag the sizer block to a new position.

   As you drag the sizer block the object is extended (or compressed) in the direction in which you drag.
4. When the object reaches the desired size release the mouse button.

Note: By default, when you resize an object it conforms to an invisible ~~grid~~ in the work area. This feature lets you align objects vertically and horizontally.

Related Topics:

Hints for Resizing Text Objects
Hints for Resizing Graphic and OLE Objects

Resizing Text Objects

Often, you'll find it useful to resize a text object. For example, you may create a text object, begin entering text in it, and find that the object isn't large enough to hold all the characters you want. When you make a text object larger you are creating more space for entering additional text.

When you need to make a text object smaller you have to be careful not to make it too small. For example, if you draw a text object, enter text in it, and then make the text object smaller so that it closely frames the text, you may accidentally *scroll* some characters out of view. You can bring characters back into view by making the object slightly larger and using the arrow keys.

Resizing Graphic and OLE Objects

By default, graphic and OLE objects are not resizable. You'll find that initially, when you draw one of these objects and select it, you can only move the object, not resize it. This is because of the default settings for the DrawStyle property for graphic and OLE objects. By default, the graphic object's DrawStyle is set to Size Graphic. This means that the object is automatically resized to fit the graphic. With a similar result, the OLE object's DrawStyle is set to Server Size. This means that the OLE object is automatically resized to the dimensions dictated by the server application.

You can make graphic and OLE objects resizable by changing the DrawStyle property to another value such as Scale Graphic for a graphic object or Object Size for an OLE object. As an example, when you use Scale Graphic, the graphic is scaled to fit within the object as you drew it. Keep in mind that if you use a property setting like Scale Graphic or Object Size, resizing a particular dimension of the object may result in distorting the image by stretching or compressing it.

Moving Objects

There are two basic ways to move objects.   You can drag one or more objects to another position on the page.   Or, if an object is part of a **stack** of objects, where one object appears to be in front or in back of other objects, you can move an object so that it is positioned differently in the stack.

Related Topics:

Dragging Objects
Re-Ordering Objects in a Stack

Dragging Objects to a New Position

Once you have selected one object or a group of objects, you can drag to a new position on the page. This is useful because, frequently, you will find that you did not draw an object in precisely the position you want.   When you drag an object the information within it, for example, text or a graphic, moves with it.

To drag objects:

1. Select the object or the group of objects.

   The borders of any selected objects are surrounded by eight sizer blocks, one at each corner and one in the center of each side.
2. Position the cursor over the selected object or anywhere over the selected objects.
3. Press and hold the left mouse button and drag the object(s) to the new location.
4. Release the left mouse button to re-draw the object(s) at the new location.

Note: By default, when you move text objects they conform to an invisible grid in the work area.   This lets you align objects vertically and horizontally.

Re-ordering Objects in a Stack

You can create or drag objects so that they overlap or completely hide other objects. It may be that you deliberately stack objects to create a particular pattern on the page or you may stack a group of objects to one side of your work area to create room to work with another object.

When objects overlap one object always appears to be in front of the others. The other objects are stacked behind the front one. The further back in the stack they are the closer they appear to the background.

If you want to work with an object that is completely obscured drag other objects out of the way until at least a part of the object you want becomes visible. If you want to work with an object that is partially hidden use one of the re-ordering commands to change the position of an object within a stack. You can bring an object to the front of a stack or send it to the back. Or, you can move an object forward one position or backward one position.

As an example, if you wanted to work with a text object that was completely hidden by another text object, you could select the forward-most text object and issue a command to send that object to the back of the stack. When the top text object moves to the back the hidden text object comes into view.

To change the position of an object in a stack:

1. Select the object.
2. Click on the Object Order button.
3. Choose a re-ordering command.

The following table describes the effect of each re-ordering command.

Bring Forward	Brings the currently selected object one position forward in a stack.
Send Backward	Moves the currently selected object one position backward in a stack.
Bring to Front	Brings the currently selected object to the front of a stack.
Send to Back	Sends the currently selected object to the back of a stack.

Duplicating Objects

Duplicating objects is particularly helpful when you want to create a group of objects that share one or more common characteristics. For example, perhaps they all need to be exactly the same size and belong to the same family. (All objects have a Family property. When objects belong to the same family, at runtime an object icon can be defined to act on an entire family.)

To copy an object:

1. Select the object you want to copy.
2. Click on the Copy button or press CTRL + INSERT.

The selected object is copied and stored in the Clipboard. The second step in duplicating an object is to paste it from the Clipboard back onto a page.

To paste an object:

1. Make sure the correct page is showing.

   If you want to paste the object back onto the same page, you don't have to do anything. If you want to paste it onto another page, go to another page.
2. Click on the Paste button or press SHIFT + INSERT.

The Paste button appears as follows:

A copy of the object that is stored on the Clipboard is pasted into the upper left corner of the page. The new object is also selected. As necessary, drag the object to a new position.

Removing Objects

You can cut or clear objects to remove them. When you cut an object it is removed to the Clipboard. Once it is on the Clipboard you can paste it back onto a page to recover it. In fact, you can paste it as many times as you want. When you clear an object it removes the object permanently.

Cutting Objects

When you cut a selected object from the current page a copy of the object is stored in the Clipboard. Optionally, you can paste the object back onto the page, or another page, as many times as you want. Remember that an object remains on the Clipboard until you copy or cut another object, or until you quit the SmartObject Editor.

To cut an object:

1. Select the object to be cut.
2. Click on the Cut button or press SHIFT + DEL.

   The Cut button appears as follows:

   The selected object is deleted and stored on the Clipboard. If necessary, click on the Paste button to paste the object back into the upper left corner of the page.

Clearing Objects

When you clear a selected object from the current page the object is removed permanently. Use care when you clear objects because they are not recoverable. When you clear an object, it is not stored in the paste buffer and no longer exists on the page.

To clear an object:

1. Select the object(s) to be cleared.
2. Choose the Clear command from the Edit menu.

   A default setting causes a message to appear asking you to confirm that you want to clear the selected object.
3. Choose Yes to clear the object or No to Cancel.

   When you choose Yes, the selected object(s) is deleted.

Optionally you can turn off the message that confirms the Clear action by choosing Confirm Clear from the Options menu

Setting Object Properties

Each object has several properties some of which are common to other objects. Most properties are set to a default and can be reset within the SmartObject Editor. Many can also be set at runtime via the ObjSet icon. A small number of properties can *only* be changed at runtime using this icon.

To set a property in the SmartObject Editor:

1. Click on the object with the right mouse button.

   A pop-up menu appears. This menu varies from one object to another but always begins with the Properties... option. The Properties option lets you change any of the properties associated with the object.

2. Choose Properties...

   A Properties dialog box appears. A list box identifies all of the properties available for the current object. The first property is highlighted by default. The text box at the top of the dialog box contains the value(s) currently set for the highlighted property.

3. Click on the property you want to set.

4. Change the current setting by typing a new value in the text box or by choosing an option from the drop-down list.

   The drop-down list box options vary from one object to another and from one property to another. The list options are either explicit values (such as true or false) or they display a dialog box (such as the Color dialog box) to allow you to select a value.

   As soon as you click in the text box or click on the drop-down arrow the X and √ buttons are activated. If you change your mind, before you move to another property you can click on the X to re-display the original value.

5. As necessary, when you finish working with one property you can move to another and change its setting as well.

   You can use the mouse or keyboard to select a different category. To use the mouse, click on the property you want with the left mouse button. To use the keyboard first click on the √ button to accept the setting for the current property and then use the arrow keys to scroll to another property in the list box.

6. When you are satisfied with the property settings click on Done in the Properties dialog box.

Button Objects

Buttons let you create the variety of buttons that you see in Microsoft Windows applications: push buttons, icons, check boxes, radio buttons, and group boxes.

When you right mouse click on a button object you can choose the Button Styles... option to show the following list of available button styles:

Push Button
Icon Button
Check Box
Radio Button
Group Box

After you select the desired style, choose OK to close the dialog box. When you subsequently open the Properties dialog box, you will find that the selection of properties varies depending on the style of button you have chosen.

The following list shows all of the button object properties. Some of these properties do not appear in the object's property dialog box because they can only be manipulated at runtime.

Area
Bottom
Checked
CheckedRadioButton
ColorBackground
ColorText
CursorName
DeleteProtected
Enabled
FamilyName
Focus
Font
Height
IconFileName
KeyboardTabStop
Label
Left
Location
NotifyOnClickLeft
NotifyOnClickMiddle
NotifyOnClickRight
ObjectName
Rectangle
Right
Size
Style
Top
Visible
Width

Push Button

The default is the conventional Windows style push button that you would use for an OK or Cancel button.

Icon Button

Icon buttons are similar to push buttons, but instead of a text label they contain an image. The image is loaded into the button when you use the IconFilename property to specify an .ico file.

Check Box

Users can toggle check boxes on and off by clicking on them. Check boxes work independently of each other. Example: If you present users with three check boxes, they can turn any number of them on or off. If you want a check box to be checked (marked with an X) by default, set its Checked property to True. At runtime, use an ObjGet icon to learn whether a user turned a check box on or off. The ObjGet icon can retrieve the current setting of the object's Checked property. Your application can branch accordingly.

Radio Button

Like check boxes, radio buttons toggle on and off when a user clicks on them. Unlike check boxes, radio buttons act as a group because the user cannot select more than one option.

The way to identify radio buttons as a group is to assign them the same FamilyName property value. At runtime, after a user selects a radio button option, your application can use an ObjGet icon to learn which option a user chose. In the ObjGet icon, specify Family as the Scope. Specify the FamilyName of the radio buttons in the Name text box. Specify CheckedRadioButton as the Property. Specify a variable in which you want to store the object name of the radio button that was selected. Use a Branches composite to evaluate the name of the chosen object and branch accordingly.

Group Box

Use a group box to logically organize other buttons. The group box is different from the other button styles because users don't actually select it. Rather, it is a tool for organizing other buttons.

Graphic Objects

Graphic objects let you display graphics on a page. The graphics can be actual size, stretched or compressed, or tiled within a particular area. Different properties are available for a graphic object depending on whether the object is live or static.

Related Topics:

Creating Objects
Making an Object Live or Static
Creating Input Selectable Objects

The following list shows all of the graphic object properties. Some of these properties do not appear in the object's property dialog box because they can only be manipulated at runtime.

Area
Border
Bottom
CursorName
DeleteProtected
DrawStyle
FamilyName
FileName
Height
Left
Location
NotifyOnClickLeft
NotifyOnClickMiddle
NotifyOnClickRight
ObjectName
Rectangle
Right
SelectionArea
Size
Top
Visible
Width

Listbox Objects

Users can make selections from list boxes. If the choices that you provide do not fit in the list box, the list box automatically has a scroll bar. You decide the data, the font, and whether you want the items to be organized alphabetically.

Related Topics:

Creating Objects
Entering Items into Listboxes

The following list shows all of the listbox object properties. Some of these properties do not appear in the object's property dialog box because they can only be manipulated at runtime.

Area
Bottom
ColorBackground
ColorText
CursorName
DeleteProtected
Enabled
FamilyName
Focus
Font
Height
ItemList
KeyboardTabStop
Left
Location
NotifyOnSelect
NotifyOnSelectChange
ObjectName
Rectangle
Right
SelectedItemData
SelectedItemNumber
ShowPartialItems
Size
Sort
Top
Visible
Width

Entering Items into Listboxes

When you right mouse click on a listbox object the pop-up menu contains an Item List... option. Use this option to specify the items you want to appear in the List. When you choose Item List... the Item List dialog box appears. Type an item into the top text box and click on Add to add the item to the list.

Each time you type an item and click on Add the item is appended to the end of the list. If you click on an item in the list and then choose Add, the item you typed in the top text box is added to the list just above the currently selected item. When you want to append to the end of the list again, choose Clear to remove the highlight from the list.

You can also move an item from one position in the list to another. Click on the item you want to move. The item is highlighted. Move the mouse cursor to the location in the list where you the item to move. Click the left mouse button to move the item.

To remove an item click on the item and choose Delete. To remove all items, choose Delete All. If you change your mind choose Cancel. The dialog box is removed without putting any of your changes into effect. When you are finished with the dialog box and want to accept the changes you have made, choose Done. (The items in the Item List dialog box will be sorted alphabetically if you set the listbox's Sort property to True.)

OLE Objects

OLE objects take advantage of the Microsoft Windows OLE (Object Linking and Embedding) feature. From within the SmartObject Editor you can access any Microsoft Windows application that is OLE-ready and create data that eventually becomes an object within your IconAuthor application. For example, you can draw an OLE object, and through it you can access Microsoft Graph to create a chart. When you exit the server application (Microsoft Graph) the chart becomes an object on the SmartObject page.

Once you use a server to create data for an OLE object, you decide the **action** that is used by that particular object at runtime. This is the action that the object takes when a user double clicks on it at runtime. One option is to use the action that is the default as designated by the server. For example, the server-designated action for Microsoft Draw graphic data is *edit*. The edit action means that when a user double clicks on the object at runtime, Draw appears and the user can edit the graphic. As another example, the server-designated action for an audio wave file is *play*. When a user double clicks on the object at runtime, the audio wave file plays.

In order for the play or edit actions to work, the server application must be present. The SmartObject Editor lets you set a property called DefaultAction that specifies whether the object should use the server-designate action, an alternative action, or whether the object should do nothing when the user double clicks on it. (This last option is obviously effective for an OLE object with data such as a chart or a graphic, but not for data such as a wave file.)

When you right mouse click on an OLE object the following commands are available:

Insert New Object...
Convert Data To Static
Paste
Paste Links
Links...

Depending on how you set certain properties for an OLE object, you can even allow users to access the server application at runtime (by double clicking on the object), so that they too can edit the object. Different properties are available for an OLE object depending on whether the object is live or static.

Related Topics:

Creating Objects
Making an Object Live or Static
Creating Input Selectable Objects

The following list shows all of the OLE object properties. Some of these properties do not appear in the object's property dialog box because they can only be manipulated at runtime.

Area
Bottom
ColorBackground
CursorName
DefaultAction
DeleteProtected
DrawStyle
FamilyName
Height
Left
Location
NotifyOnComplete
NotifyOnStart
ObjectName
Rectangle

Right
SelectionArea
Size
State
Top
Visible
Width

Insert New Object... Command

This command displays the Insert New Object dialog box.

The Object Type list box shows the registered OLE servers available on your system. When you choose a server and click on OK, the application runs and you can create the data (for example a spreadsheet or a sound file. Within the server, choose Update from the File menu before you exit. Then choose Exit & Return to IA Object #xxxx from the File menu. The data you created is displayed in the OLE object on the page.

Convert Data To Static Command

This command converts the *data* in an OLE object to static. When the data is static it cannot be changed because the connection to the server has been permanently severed. Static data is different from a static object (an object that is not live). An OLE object with static data can be live and still have its properties, such as visible and enabled, changed at runtime.

Paste Command

This command lets you paste data (such as a picture) that was previously cut or copied to the Clipboard (from a server application) into your OLE object.

Paste Link Command

This command lets you paste server data from the Clipboard in a manner similar to the previously described Paste command. However, when you use the Paste Link command, the data is linked to the file from which it was originally copied. As a result, any time that you open the server application and change and save that file, the changes you make are automatically updated in the OLE object on the SmartObject page.

Links... Command

This command is available if you right mouse click on an OLE object on which you have previously used the Paste Link command. The Links... command displays the Links dialog box that describes the current status of the link, such as the server name and filename. This dialog box lets you edit the link information. Refer to your Microsoft Windows documentation for more information on changing link information.

Text Objects

Text objects let you display text on a page. This can be text that you provide for display purposes or it can be text input by the user at runtime. If a text object is intended for user input, after the user enters the text, an ObjGet icon can retrieve the string that the user entered and store it in a variable. Once the string is in a variable, your structure can manipulate the string. For example, a Write icon can re-display the string at some later time, or a Branches composite can evaluate the string and branch to a different group of icons depending on the input.

A text object can contain any quantity of text: a character, a word, or multiple paragraphs. If all the text doesn't fit, the text object can have a scroll bar so that users can move through the text as necessary. In addition to setting properties for the general characteristics of the text object, you can also set properties for the appearance of text within the text object.

Related Topics:

Making an Object Live or Static
Making a Text Object Editable
Setting an Input Style
Creating Input Selectable Objects
Working with Text in a Text Object
Working with the Style of a Text Object

The following list shows all of the text object properties. Some of these properties do not appear in the object's property dialog box because they can only be manipulated at runtime.

AlignHorizontal
Area
BaseLine
Bottom
CharacterCurrency
CharacterDecimal
CharacterFalse
CharacterThousands
CharacterTrue
ColorFace
ColorFill
ColorFrame
ColorHighlight
ColorShadow
ColorText
CursorName
DecimalPlaces
DeleteProtected
Editable
Enabled
FamilyName
FileName
Focus
Font
Height
InputLimit
InputLimitBeep
InputTerminationRequired
KeyboardTabStop
Left
LightSource

LineSpace
Location
Mask
MultipleLines
NotifyOnClickLeft
NotifyOnClickMiddle
NotifyOnClickRight
NotifyOnComplete
NotifyOnInputLimit
ObjectName
Rectangle
Right
ScrollBarVertical
SelectionArea
Size
Text
TextCase
TextLength
Top
Visible
Width
WidthEdge
WidthFrame

Making an Object Live or Static

The most important property of an object is whether it is **static** or **live**. If an object is static it cannot be changed at runtime. If an object is live, it can be changed at runtime via a group of object icons in IconAuthor.

By default, objects are live. Only a graphic, text, and OLE objects have the option to be live or static. If, as an example, a graphic is a live object, you could have a user click on the graphic and change the property of the graphic so that a different image appears in its place. Often, however, you will find it useful to make graphic objects static. By making a graphic static it's as if the graphic becomes part of the unchanging background of the page.

Using live or static objects affects the way in which you can make your application interactive. If you use live objects, the user can manipulate them at runtime and you can use IconAuthor's object icons to change the objects' properties. If you use static objects, you can make your applications interactive via Input Selectability. When part of a page (an object for example) is input selectable, it is a hotspot that a user can select, for example, by clicking on it.

To make an object live or static:

1. Click the right mouse button on the object.

   A pop-up menu appears.
2. Choose the Object Type... command.

   The Object Type dialog box appears.
3. Choose the appropriate option, live or static, and choose OK.

When you select the desired option, choose OK to close the dialog box. Eventually, when you open the Properties dialog box, you will find that the selection of properties varies depending on whether the object is live or static.

Related Topics:

Input Selectability
Live Object Maintenance

Setting an Input Style

The term "input style" refers to text that a user inputs as well as text that the author inputs. When you set the input style you are controlling the kind of characters that can be entered in that text object and in some circumstances, the positioning and quantity of text allowable. As an example, you can set the input style to Numeric so that the user can only enter numbers in that text object.

When you right mouse click on a text object, a pop-up menu appears. Choose Input Styles... to display the Input Styles dialog box.

Select the desired option, choose OK to close the dialog box. When you subsequently open the Properties dialog box, you will find that the selection of properties varies depending on the style you selected.

The following input styles are available.

Standard
Alphabetic
Alphanumeric
Currency
Financial
Fixed Decimal
Logical
Numeric
Picture

Standard

This is the default style. It allows any characters and symbols and uses the TextCase and MultipleLines properties.

Alphabetic

This is almost identical to the Standard style but allows only alphabetical characters and spaces.

Alphanumeric

This is almost identical to the Standard style but allows only alphabetical characters, numeric characters, and spaces.

Currency

Allows numbers only and uses the following properties:

CharacterCurrency
CharacterDecimal
CharacterThousands
DecimalPlaces
Baseline

The following example shows how input might appear in this style:

$320,000.00

Financial

Allows numbers only and uses the following properties:

CharacterDecimal
CharacterThousands
DecimalPlaces
Baseline

This property is almost identical to the Currency style except that it does not use the CharacterCurrency property. The following example shows how input might appear in this style:

320,000.00

Fixed Decimal

Allows numbers only and uses the following properties:

CharacterDecimal
DecimalPlaces
Baseline

This property is almost identical to the Currency style except that it does not use the CharacterCurrency and CharacterThousands properties. The following example shows how input might appear in this style:

320000.00

Logical

Allows only one of two characters.   You set the two possible characters via the CharacterFalse and CharacterTrue properties.   By default, CharacterFalse is set to "F" and CharacterTrue is "T."

Numeric

Allows numbers only with two exceptions: the first character can be a + or - sign *and* the user can enter a decimal point (just one) at any position in the number. With the Numeric style you can use paragraph alignment to control whether input appears on the screen from left to right (make it a left-aligned paragraph) or right to left (make it a right-aligned paragraph). Note that if you use this style, the input cannot be on multiple lines.

Picture

When you choose this style, you are indicating that you want to create a mask that strictly controls which kinds of characters a user can enter and how many.   If you choose this style, the Mask property becomes available.

Making a Text Object Editable

By setting the Editable property of a live text object you can make it **editable**, that is, it acts as a text box in which the user can type. The text that is input is assigned to a property called Text. Using an ObjGet icon at runtime, the text can be stored in a variable and further manipulated or evaluated.

Timer Objects

Timers let you use time to control your application. For example, a timer can count up from 0 to measure how long it takes a user to perform a task. To understand timers you must understand the concept of **events**. An event is an action (performed by a live object) that is recognized by your application. For example, when a timer count down reaches 0, that is also an event.

When an event occurs the name of the object that generated the event is stored in the system variable @_Object_Name and the name of the event is stored in the system variable @_Object_Event. (When a timer object event occurs, the string "alarm" is stored in @_Object_Event.)

You can set up your application so that when a timer event occurs, execution flows in a particular direction based on the event.

Click the right mouse button on a timer object to display the Timer Styles... option. The following styles are available:

Periodic
Count Down
Count Up
Alarm

Related Topics:

Creating Objects

The following list shows all of the timer object properties. Some of these properties do not appear in the object's property dialog box because they can only be manipulated at runtime.

Area
Bottom
DeleteProtected
Enabled
FamilyName
Height
Left
Location
ObjectName
Rectangle
Right
Size
Style
TimerData
Top
Width

Periodic Timer Style

Timer causes an event to occur every x seconds. You set the period length via the TimerData property. The TimerData should be expressed in hh:mm:ss for example, 2:30:00 would cause an alarm event to occur every 2 and 1/2 hours and 20 would cause an alarm event to occur every 20 seconds. Each time the timer reaches 0 it resets itself.

Count Down Timer Style

Timer gives the user a fixed amount of time to perform a task. The TimerData should be expressed in hh:mm:ss. For example, 5 gives the user 5 seconds, 5:0 gives the user 5 minutes, and 5:0:0 gives the user 5 hours.

Count Up Timer Style

Timer lets you determine how long it took a user to do a task.   In this case, you would set the TimerData to start at 0.   When an event occurs, such as a user clicking OK to move on to the next question in a quiz, you can use an ObjGet icon to retrieve the TimerData property of the Timer object.   The TimerData property will tell you the time in seconds.   For example, the value 120 equals 120 seconds or 2 minutes.

Alarm Timer Style

Timer causes an alarm event to occur at a specific time of day. Use the TimerData property to set the time at which you want to the alarm event to occur. Be sure to express the TimerData in military time. If you want a kiosk to start up every day at 8 am, you could set the TimerData to 08:00:00 and set the Style to Alarm.

Transparent Objects

A Transparent object acts like a push button that cannot be seen. The user can click on it and cause an event to occur. If for example, you draw a Transparent object that covers the entire page, the user will not see a button but will be able to click anywhere to cause an event to occur.

One important note about the Transparent object is that like many other objects, it too has a Visible property. Remember that for button objects, not only does the Visible property remove the button from view, but it also makes it so the user cannot click on it. The Transparent object has a Visible property so that you can disable it simultaneously with other non-transparent objects. For example, your application can have a family of objects called ColorButtons. The ColorButtons family is made up of three push buttons and a transparent object. Since they all have the same FamilyName, you can change the Visible property of the ColorButtons family to false and remove from view/disable the entire family at once.

Also, when you draw a Transparent object, be sure not to have any other live objects on the page completely overlap it. Any part of the Transparent object that is covered by another live object cannot be activated at runtime.

The following list shows all of the transparent object properties. Some of these properties do not appear in the object's property dialog box because they can only be manipulated at runtime.

Area
Bottom
CursorName
DeleteProtected
Enabled
FamilyName
Height
Left
Location
NotifyOnClickLeft
NotifyOnClickMiddle
NotifyOnClickRight
ObjectName
Rectangle
Right
Size
Top
Visible
Width

Working with the Style of a Text Object

The style of a text object is what causes it to have certain characteristics such as its color or the width of its border. There are two basic ways to manipulate the style of a text object. You can work with the style by editing it within the SmartObject Editor, or you can work with it by changing certain specific properties at runtime (using an ObjSet icon).

Every text object has a style. When you begin drawing text objects and they are white-filled with a thin black border it's because they have a style attached to them called Default. Whenever you use the text object tool to draw new text objects they use the Default style.

Related Topics:

Changing Block Style
Creating a New Named Style
Removing a Style

Changing Block Style

To select a block style:

1. Click on a text object with the right mouse button.
2. Choose Block Styles... from the pop-up menu.

   The Block Styles dialog box appears.
3. Choose a style from the Styles list.

   Optionally, click on the Custom button to extend the dialog box and change one of the many characteristics of the object to customize it.
4. Choose OK.

Related Topics:

Creating a New Named Style
Removing a Style

Creating a New Named Style

You can create and save a style that you want to use repeatedly, adding it to the list of frequently used styles in the Block Styles dialog box. From that point you simply choose that style from the list whenever you want to use it.

To create a new style:

1. Use the extended Block Styles dialog box to create a style.
2. Choose Save...

   The Save Style As dialog box appears.
3. Type the name of the style in the Style Name dialog box and choose OK.

   The style name appears in the Styles list box. Each time you add a style it is inserted into the list in alphabetical order.

Note: You can create your own default text object style. Click on an existing style in the Styles list and click on the Set As Default button. On subsequent occasions when you start the SmartObject Editor the new Default style is used automatically.

Removing a Named Style

If you do not plan to use a style you can remove it from the list in the Styles box. Use the remove style feature carefully. When you remove a style it is un-retrievable and must be recreated to be re-included in the list.

To remove a style from the Styles area:

1. Click on the style name to highlight it.
2. Choose Remove.

   The style is removed from the list.

Working with Text in a Text Object

After you draw a text object you can put text in it. Use one of the following methods:

Type and edit text via the keyboard
Create a text file and specify it as the Filename property of the object.

Entering Text

The text objects you create can contain any quantity of text. For example, a text object can be empty, contain a single character, or multiple paragraphs.

To begin entering or editing text double click on any text object. The mouse pointer changes to an I-beam cursor and a flashing vertical caret called the **insertion pointer** appears within the text object. If the text object you double click in does not yet contain text the insertion pointer is automatically placed in the top left corner. If the text object already contains text the insertion pointer is placed within the text at the position where you double clicked.

After you place the insertion pointer in a text object you can begin typing. As you type the characters appear on the screen and the insertion pointer moves to the right. In some situations, a text object may require only a word or phrase such as "Help" or "Exit." In other cases text objects will contain multiple paragraphs of text.

The width of a line of text is defined by the left and right margins of the text object. The amount of text you can put in a text object is unlimited. When the insertion pointer reaches the right margin it "wraps around" or jumps back to the beginning of the next line. Press ENTER at any time to start a new line or skip one or more lines. If you make a mistake you can use one of the editing techniques described in subsequent sections or you can simply press the BACK SPACE key to remove text one character at a time.

Related Topics:

Moving the Insertion Pointer
Scrolling Text
Typing IconAuthor Variables
Editing Text
Editing Paragraph Alignment and Spacing
Changing Fonts
Changing Text Color
Working with Text Styles

Moving the Insertion Pointer

If you decide to go back and add a word or sentence to the middle of a line or paragraph you can move the insertion pointer to the desired position and begin typing. There are two ways to move the insertion pointer. You can use the arrow keys or click the I-beam shaped mouse pointer at the desired position in the text. If you use the mouse technique the insertion pointer is placed wherever you click.

By default, when you first double click in a text object Insert mode is in effect. When you type in Insert mode characters are added to the body of text without affecting any existing characters. The alternative mode is Overstrike. Use Overstrike when you want to type and replace a character for every character that you type. To change to Overstrike mode press the Insert key. The current mode is represented by the characters INS (for Insert) or OVR (for Overstrike) in the far right cell of the status bar.

Scrolling Text

When you reach the bottom of the text object the body of text scrolls upward and the insertion pointer wraps around to the beginning of the next line. Each time the body of text scrolls upward the top line moves out of the window and is no longer visible.

Text that scrolls out of view still exists and can be easily brought back into view by scrolling. Use the up and down arrow keys on the keyboard to scroll up and down. For example, if you want to see the top of the body of text, press the up arrow key. Each time you press the up arrow key, the insertion pointer moves up one line. When the insertion pointer reaches the top line of text that is visible, it stops moving and the text begins to scroll down into view.

Note: If you want a user to be able to scroll through the text in a text object at runtime you must set the object's ScrollBarVertical property to true.

Typing IconAuthor Variables

In addition to conventional text, you can type IconAuthor variable names in a text object. At runtime, your IconAuthor application displays the value stored in the variable rather than the variable name.

Any variable used in a SmartObject file must first be properly defined within the IconAuthor application.

Naming Variables

Variable names can contain up to 19 characters (including the '@' symbol). They can be made up of any alphanumeric characters (A...Z, a...z, 0...9), the underscore character '_' and matching square brackets. Matching square brackets are used to indicate indexed variables. For every occurrence of a '[' there must be a ']'. Variable names cannot include any punctuation.

Rules for Expressing Variables

1. Any non-valid character, for example, a space, or a colon (:), terminates the variable name. The space causes a space to appear on the screen directly following the value; the colon causes a colon to appear on the screen.
2. When you want a displayable character to appear immediately following the variable, use a "\" to delimit where the variable ends. When IconAuthor finds a "\" immediately following a variable name, it signals the end of the variable name and is stripped from the output. If you use a "\" to terminate a variable name when it is not necessary, it is still stripped and no errors result.
3. To use "@" or "\" symbols as literals in the SmartObject file they do not require any special treatment unless they are followed by a valid variable character.
4. If a variable name is valid, but IconAuthor cannot find a value for it, a null character (a space) is displayed.

Editing Text

There are several ways to edit text once you've entered it in a text object. You can drag selected text and drop it elsewhere in a text object or in another text object. You can cut or copy selected text to the Windows Clipboard and then paste it into any text object, even one on a different page. Because the Windows Clipboard is shared by many applications, for example, Windows Notepad, you can take the text that you copy or cut to the Clipboard and paste it into a file created with some other text editor.

Related Topics:

Selecting Text
Dragging Text
Removing Text
Copying Text
Pasting Text

Selecting Text

Many text editing tasks require that you first select the portion of text you want to change. You can select any quantity of consecutively occurring text, for example, a character, a word, or several paragraphs.

To select text:

1. Position the I-beam mouse pointer over the first character in the range you want to select.
2. Press and hold the left mouse button and drag the mouse cursor to the last character in the range you want to select.

   As you drag the mouse cursor, the text you select is highlighted.
3. Release the left mouse button.

   The text is now selected and can be altered.

Hints for Selecting Text

The following are some shortcuts for selecting text:

Double-click on a word to select it.

Double-click immediately after a word and select that word and the next word.

Click to position the insertion pointer, hold down the Shift key, and click at another point further down in the text. This selects all text between the two positions .

Click to position the insertion pointer, hold down the Shift key, and press the arrow right key. This selects one character at a time.

Dragging Text

You can drag selected text from one position to another. This can be done within one text object or from one text object to another.

To drag text:

1. Select the text you want to move.
2. Position the I-beam anywhere over the selected text.
3. Press and hold the left mouse button and drag the selected text to the new location (in the same text object or in another text object).

   While you are moving the mouse the appearance of the cursor varies. Whenever the cursor is over a valid drop position (a text object) it appears as an I-beam with a T below and to the right of it. Whenever the cursor is over an invalid drop position (not over a text object) it appears as a circle with a bar through it.
4. When the cursor is over the proper drop position release the left mouse button.

   The text is moved from the original location to the new location.

Note: To copy the text from one position to another instead of moving it, press and hold the CTRL key before you press and hold the left mouse button. Continue to hold the CTRL key until you have dropped the copied text.

Removing Text

One way to remove text is one character at a time. Each time you press the backspace key one character to the left of the cursor is removed. Each time you press the Delete key one character to the right of the cursor is removed. You can also replace any amount of text by selecting it, and then typing new text. When you begin typing, the selected text is removed.

The SmartObject Editor also lets you use the Cut or Clear text with buttons or commands from the Edit menu.

Cutting Text

When you cut text it is stored on the Windows Clipboard. The next text that you cut or copy to the Clipboard replaces the item that is currently stored there.

To cut text:

1. Select the text to be cut.
2. Click on the Cut button.

   Note: Choosing the Cut command from the Edit menu or pressing SHIFT + DEL is the same as clicking the Cut button.

   The selected text is deleted and stored on the Clipboard.

Clearing Text

When you clear text it is not stored on the Clipboard and it no longer exists on the page.   Text acted upon with the Clear command is not recoverable.

To clear text:

1. Select the text to be cleared.
2. Choose the Clear command or press DEL.

   The selected text is deleted.

Copying Text

There are two basic ways to copy selected text. You can use the Copy button. You can also press and hold the Ctrl key while you drag selected text to a new location. When you copy text its appearance on the page is unchanged, but a copy of the item is stored on the Windows Clipboard. The next text that you copy or cut to the Clipboard replaces the item that is currently stored there.

To copy text with the Copy button:

1. Select the text to be copied.
2. Click on the Copy button.

   Note: Choosing the Copy command from the Edit menu or pressing CTRL + INSERT is the same as clicking on the Copy button.

   The selected text is copied and stored on the Clipboard.

To copy text by dragging it:

1. Select the text to be copied.
2. Position the I-beam anywhere over the selected text.
3. Press and hold the CTRL key.
4. Press and hold the left mouse button and drag the selected text the new location (in the same text object or in another text object).

   While you are moving the mouse the appearance of the cursor varies. Whenever the cursor is over a valid drop position (a text object) it appears as an I-beam with a + sign with a T below it.

   Whenever the cursor is over an invalid drop position (not over a text object) it appears as a circle with a bar through it.
5. When the cursor is over the proper drop position release the left mouse button.

   The text is copied to the new location. A copy of the text is also placed on the Windows Clipboard. You can paste the text from the Clipboard as many times as you want.

Note: To move the text from one position to another instead of copying it, *do not* press and hold the CTRL key before you press and hold the left mouse button.

Pasting Text

Use the Paste button to paste a copy of text that is currently stored on the Clipboard into a text object. When you paste text onto a page a copy of the item still exists on the Clipboard.

Note: In order to paste text the last item cut or copied to the Clipboard must have been text. If the Clipboard is empty or contains some other kind of item, you cannot paste text.

To paste text:

1. Position the insertion pointer at the location where you want to insert the text.
2. Click on the Paste button.

   Note: Choosing the Paste command from the Edit menu or pressing SHIFT + INSERT is the same as clicking on the Paste button.

   A copy of the text stored on the Clipboard is pasted into the text object at the point where you placed the insertion pointer. You may paste the same text in other text objects until you do the next cut or copy.

Editing Paragraph Alignmenu and Spacing

By default a paragraph is left-aligned and single-spaced. You can change the alignment and line spacing of a paragraph before you enter text in it, or after.

To change the paragraph style:

1. Position the text insertion pointer in the paragraph you want to change.
2. Right Mouse click anywhere in the text object to display a pop-up menu.
3. Choose one of the available options.

Left Text is aligned along the left border of the text object.

Center Text is centered within the text object.

Right Text is aligned along the right border of the text object.

Single There is no extra space between successive lines of text.

1 1/2 A space that is half the height of the current line, is drawn between successive lines of text.

Double A space that is the height of the current line is drawn between successive lines of text.

Changing Fonts

You can change the font of selected text or text that you are about to type.

To change the font:

1. Position the text insertion pointer at the point you want to begin using a new font or select a portion of text.
2. Right mouse click on the text.

   A pop-up menu appears.
3. Choose Fonts... from the menu.

   The Font dialog box appears.
4. Make changes as desired, for example, to the font, font size, and font color.
5. Choose OK.

When you choose OK in the Font dialog box the changes you made are reflected in the text you type, or in the text that is selected.

Changing Text Colors

By default, text is black with a white background (fill color). You can change both the text color and the fill color.

To change colors:

1. Position the text insertion pointer at the point you want to begin using a new color or select a portion of text.
2. Right mouse click on the text.

   A pop-up menu appears.
3. Choose Text Color... or Fill Color... from the menu.

   A Text or Fill Color dialog box appears.
4. Choose a color from the Color drop-down list box and choose OK.

When you choose OK the changes you made are reflected in the text you type, or in the text that is selected. Be aware, however, that when text is selected, its colors appear reverse highlighted. For example, red appears as green. When you de-select the text the correct colors will appear.

Working with Text Styles

When you change the style of text you alter one or more of the following characteristics:

font type
font size
font style (bold, underlined, or italic)
text color
text fill color

For example, text in a text object can use any number of the available font types. You can type a title using one font and then change the font characteristics and continue typing several other paragraphs. You can then select a sentence in the middle of one of the paragraphs and change its font characteristics, leaving the surrounding text unchanged.

To select a text styles dialog box:

1. Make sure you are editing text.

   Either the text insertion pointer must be in a text object or a portion of text must be selected.
2. Right mouse click on the text object.
3. Choose Text Styles...

   The Text Styles dialog box appears.
4. Click on a style to select it.

   Optionally, click on the Custom button to alter the font or fill color of the text.
5. Choose OK.

Related Topics:

Creating a New Named Style
Removing a Named Style

Creating a New Named Style

You can create and save a style that you want to use repeatedly, adding it to the list of frequently used styles in the Text Styles dialog box. From that point you simply choose that style from the list whenever you want to use it.

To create a new style:

1. Open the Text Styles dialog box and click on the Custom>> button.
2. Use the Font dialog box, the Text Color dialog box, and/or the Fill Color dialog box to change the current settings.
2. Choose Save...

   The Save Style As dialog box appears.
3. Type the name of the style in the Style Name dialog box and choose OK.

   The style name appears in the Styles list box. Each time you add a style it is inserted into the list in alphabetical order.

Note: You can create your own default text object style. Click on an existing style in the Styles list and click on the Set As Default button. On subsequent occasions when you start the SmartObject Editor the new Default style is used automatically.

Removing a Named Style

If you do not plan to use a style you can remove it from the list in the Styles box. Use the remove style feature carefully. When you remove a style it is un-retrievable and must be recreated to be re-included in the list.

To remove a style from the Styles area:

1. Click on the style name to highlight it.
2. Choose Remove.

   The style is removed from the list.

Creating Text Files

Text objects are the means by which text appears on your SmartObject page. The simplest way to display text is to type it directly into the text object. However, it is also possible to create text in a separate file and then display the file within the text object at runtime.

Text objects have a property called Filename. If you create a text object and set its Filename property to the name of a text file, the contents of the text file is displayed in the object at runtime. Do not type text in a text object that has a text filename associated with it. The text that you type directly into the object will not appear at runtime.

One way to create text files is to work in a different mode within the SmartObject Editor. Simply by choosing New... from the File menu you can change to Formatted Text or ASCII Text mode and begin creating files. Formatted Text mode means that the characters can be formatted (for example, in terms of color, font and size). Text files created in ASCII Text mode have no character formatting.

Another way to create text files is to use any ASCII text editor (such as Notepad) or any Word Processor that supports .RTF (rich text format) files (such as Microsoft Word for Windows).

Text files can be an exceptionally useful way to include text in a text object.

Using Formatted Text Mode

Use the SmartObject Editor in Formatted Text mode to create character-formatted text files that can be linked or embedded in text objects via the Filename property.

When you use Formatted Text mode to create a text file you are really creating the contents of a text object. Appropriately, in terms of entering and editing text, Formatted Text mode follows the basic rules for working with text in text objects. That is, entering text (including IconAuthor variables), and copying, cutting, and pasting text all work the same as they do for a text object in Page Layout mode.

To change to Formatted Text mode:

1. Choose New... from the File menu.

   The New... dialog box appears.

2. Click on the Formatted Text option and choose OK.

A blank, untitled file appears. Also, the ribbon bar disappears and the SmartObject Editor menu items change. The menus now appear as File, Edit, Text, and Help.

Formatted Text Mode - Edit Menu

Cut

Deletes the selected text.

Copy

Copies the selected text onto the Clipboard.

Paste

Pastes text into the file at the current position of the cursor.

Clear

Clears the selected text.   Cleared text is permanently removed and is not placed on the Clipboard.

Select All

Selects all text in the file.

Formatted Text Mode - Text Menu

Fonts...

Accesses the Font dialog box which allows you to define the font characteristics of the text.

Text Color...

Shows the Text Color dialog box that allows you to select a solid color for text.

Fill Color...

Shows the Fill Color dialog box that allows you to select a color for the background of text.

Text Styles...

Shows the Text Styles dialog box that allows you to select or edit a style, create a new style, or remove a style.

Left

Left-aligns text within the current paragraph.

Center

Centers text within the current paragraph.

Right

Right-aligns text within the current paragraph.

Single

Single spaces text within the current paragraph.

1 1/2

1 1/2 spaces text within the current paragraph.

Double

Double spaces text within the current paragraph.

Formatted Text Mode - File Menu

Formatted Text mode lets you create text files with in the .FTT format. You can start completely new files in this mode; or you can open an .RTF file created with any Word Processor or editor that supports .RTF. You can also open an .smt file or a .TXT file, but doing so changes the mode to Page Layout or Text mode respectively.

You can save your files as .FTT files. .FTT, .RTF and .TXT file formats can be linked to or embedded in a text object via the Filename property. Also, an .RTF file can be opened in any Word Processor or text editor that supports .RTF.

New...

Displays the New dialog box which lets you choose to open a new SmartObject file in one of the three modes: Page Layout, Formatted Text, or Text.

Open...

Lets you use an Open dialog box to open an existing file. The file you choose to open can be in Page Layout format (.SMT), in Formatted Text format (.FTT), in ASCII text format (.TXT) or in Rich Text Format (RTF.). Depending on which kind of file you choose to open, the SmartObject Editor comes up in the appropriate mode. If you open an .RTF file the editor stays in Formatted Text mode.

Save

Lets you save the current file.

Save As...

Lets you use a Save As dialog box to save the current file as an .FTT file.

Exit

Exits the SmartObject Editor.

Using ASCII Text Mode

Use the SmartObject Editor in Text mode to create ASCII text files that can be linked or embedded in text objects via the Filename property.

To change to Text mode:

1. Choose New... from the File menu.

   The New... dialog box appears.

2. Click on the Text option and choose OK.

A blank, untitled file appears. Also, the ribbon bar disappears and the SmartObject Editor menu items change. The menus now appear as File Edit, and Help.

Entering and editing text in Text mode is similar to using a text editor such as Windows Notepad. Type to enter text. Use the Backspace and Delete keys to remove characters one at a time. Click at a different point in the body of text if you want to reposition the flashing I-beam insertion pointer.

ASCII Text Mode - Edit Menu

To use the Edit menu commands, first drag across the text you want to edit to select (highlight) it. Once the text is selected you can use any of the Edit commands.

Cut

Deletes the selected text.

Copy

Copies the selected text onto the Clipboard.

Paste

Pastes text into the file at the current position of the cursor.

Clear

Clears the selected text. Cleared text is permanently removed and is not placed on the Clipboard.

Select All

Selects all text in the file.

ASCII Text Mode - File Menu

Text mode lets you create text files with in the ASCII .TXT format. You can create and save files to serve many purposes within IconAuthor. For example, .TXT files for displaying text, .VAR files for loading variables, and .fmt files for formatting new database files. Once created, a .TXT file can be linked to or embedded in a text object via the Filename property. Also, a .TXT file can be opened in any Word Processor or text editor that supports .TXT.

New...

Displays the New dialog box which lets you choose to open a new SmartObject file in one of the three modes: Page Layout, Formatted Text, or Text

Open...

Lets you use an Open dialog box to open an existing file. The file you choose to open can be in Page Layout format (.SMT), in Formatted Text format (.FTT), in ASCII text format (.TXT) or in Rich Text Format (RTF.). Depending on which kind of file you choose to open, the SmartObject Editor comes up in the appropriate mode. If you open an ASCII file the editor stays in Text mode.

Save

Lets you save the current file.

Save As...

Lets you use a Save As dialog box to save the current file as a .TXT file

Exit

Exits the SmartObject Editor.

Displaying Video in the SmartObject Editor

The SmartObject Editor allows you to work in video overlay mode in order to display the current video frame.   This helps you simulate how the page will appear in conjunction with video at runtime.

To work in video overlay mode:

- Click on the Video Overlay button in the ribbon bar.

  The current video frame (on your video player) is visible in the SmartObject Editor wherever you have used the transparent color.   Frequently the transparent color is black, but it depends on the video overlay board installed on your system.   Refer to the documentation for your overlay board to determine which color is transparent.

  Note: Video Overlay mode is only available if you have installed and configured the necessary video setup.

Creating Input Selectable Objects

SmartObject files can serve different purposes within IconAuthor applications. Frequently they are straight text and graphic displays that the user is simply intended to view. However, SmartObject files can also play an important part in creating input selectable objects from which a user can make a selection or choose multiple choice answers to a question.

The first step in creating an input selectable menu is to construct the SmartObject Page. Draw **static** text, graphic, and OLE objects so that they include the selections the user can choose from and the instructions to tell the user how to use the screen.

There are two basic ways to make a display input selectable. You can use IconAuthor's Input or InputMenu icon, or the SmartObject Editor to designate parts of the screen as input selectable.

The method that is recommended depends on the number and variety of menus you plan to display within one loop. If you plan to display a menu where the size, number, and location of the selectable areas *are the same* each time the loop is executed, designate input selectability in the IconAuthor Input or InputMenu icon. If you plan to display a menu where the size, number, and/or location of the selectable areas *varies* each time a loop is executed, designate input selectability in the SmartObject Editor.

Related Topic:

Designating Input Selectability in the SmartObject Editor

Designating Input Selectability in the SmartObject Editor

After you create a SmartObject page with only static objects, make one or more objects, input selectable. If your page contains several objects, you can make all, none, or just some of them input selectable. The objects that you make input selectable must be static.

To make an object input selectable:

1. Click the right mouse button on the object you want to make selectable.
2. Choose Object Type...
3. Click on the Static option and choose OK.
4. Click the right mouse button on the object again.
5. Choose Properties...
6. Change the SelectionArea to a unique number.

   If you leave the Area Number blank or set to 0 the object is not selectable.
7. Choose OK .

Repeat the preceding steps as necessary to designate all of the input selectable areas on the page. Note that a page can not have more than 256 input selectable areas. If you click on an object that you have made input selectable, the area number of that object is displayed in the status bar.

After you have created the SmartObject page and designated input selectable areas, access IconAuthor and build a Menu composite into your structure. When you add content to the Display icon, enter the SmartObject file and page as the filename.

When you add content to the Input Menu icon, you define *which areas on the screen you want to be input selectable.* The information that describes the location of the selectable areas goes in the Selection Areas text box. You already designated the input selectable areas within the SmartObject file, so you don't need to do it again. Enter the variable name @_TEXT_AREAS in the Selection Areas text box.

At runtime, the Display icon displays the SmartObject page on the screen, and automatically places the coordinates of the input selectable areas for that page in @_TEXT_AREAS. The Input Menu icon activates the selectable areas on the screen when it sees @_TEXT_AREAS in the Selection Areas text box. The number of the area the user selects is stored in the system variable @_SELECTION. Later, in the Choices composite, the value in @_SELECTION is evaluated and causes a particular branch to be executed.

Managing SmartObject Pages

The blank background on which you place objects is called a **page**. Each SmartObject file can contain one or more pages that are actually used as separate displays at runtime. The number of pages required by your file depends on your authoring style. Typically, you group pages together in a file because they are logically related to one another. For example, all the pages in one presentation or pageturning sequence are likely to be in one file, or all the pages associated with one part of an application or subroutine might be in one file.

When you first start the SmartObject Editor a blank page called "Untitled" is displayed. If you plan to include multiple pages in the file you will need to give each page a unique name. You also have the option of specifying a color for the page although by default pages are transparent. When you display a transparent page at runtime, the objects on that page appear to rest directly on top of any previously displayed information.

Related Topics:

Creating New Pages
Renaming Pages
Copying Pages
Deleting Pages
Importing Pages
Changing Pages
Working with Page Color

Creating New Pages

Most SmartObject files have multiple pages.

To create a new page in the current SmartObject file:

1. Click on the Page Maintenance button or choose Page Maintenance... from the Page menu.
2. Choose New.

   The New Page dialog box appears.
3. Enter the name of the new page in the Page Name text box.

   The name you use must be unique within the file and can be up to 24 characters.
4. Choose OK.

   If you specify the name of an existing page a message is displayed requesting a "unique" name. Once you have specified a unique page name the dialog box is removed and the new, blank page is displayed.
5. When you are finished with the Page Maintenance dialog box choose Close.

Note: You can also use the New Page button to display the New Page dialog box.

Renaming Pages

To rename a page in the current SmartObject file:

1. Click on the Page Maintenance button.

   The Page Maintenance dialog box appears.
2. In the Page List area click on the page you want to rename.

   The page you select appears in the work area.
3. Choose Rename.

   The Rename Page dialog box appears.

   The current name is highlighted.
4. Enter the new name in the Page Name text box.

   The new name replaces the old one. The name you use must be unique within the file and can be up to 24 characters.
5. Choose OK.
6. When you are finished with the Page Maintenance dialog box choose Close.

Copying Pages

You can copy a page. This is particularly useful if you want to create another page that looks very similar to an existing page.

To copy a page in the current SmartObject file:

1. Click on the Page Maintenance button.

   The Page Maintenance dialog box appears.
2. In the Page List area click on the page you want to copy.

   The page you select appears in the work area.
3. Choose Copy.

   The Copy Page dialog box appears.

   The current name is highlighted.
4. Enter the name of the new page in the Page Name text box.

   The name you use must be unique within the file and can be up to 24 characters.
5. Choose OK.

   The new page is displayed and appears identical to the original.
6. When you are finished with the Page Maintenance dialog box choose Close.

Deleting Pages

You can delete an entire page.

To delete a page in the current SmartObject file:

1. Click on the Page Maintenance button.

   The Page Maintenance dialog box appears.

2. In the Page List area click on the page you want to delete.

   The page you select appears in the work area.

3. Choose Delete.

   A message appears that asks if you want to delete the selected page. Choose Yes to delete or No to cancel the process.

5. When you are finished with the Page Maintenance dialog box choose Close.

Importing Pages

The SmartObject Editor allows you to import pages from other SmartObject files.

To import pages into the current SmartObject file:

1. Click on the Page Maintenance button.

   The Page Maintenance dialog box appears.
2. Choose Import...

   The Import dialog box appears.
3. Choose Browse... to identify the file from which you want to import pages.

   The Browser appears.

   Select a file and choose OK. The Browser is closed and the Import dialog box returns to view. The Import dialog box now contains the name of the file you have specified and the pages that file contains.
4. In the Pages List area, select the pages you want to import.

   To select one page, click on it.

   To select a range of pages, click on the top-most page in the range. Then, while pressing and holding the Shift key, click on the bottom-most page in the range.

   To select several pages that are not a consecutive range, for example, the first and the last page, click on one page. Then, while pressing and holding the Ctrl key, click on any subsequent pages.

   To select all the pages in the file, click on the Select All button.
5. When you have selected all the pages you want, click on Add Pages.

   The pages are appended to the end of the current SmartObject file.
6. When you are finished with the Import dialog box, choose Close.
7. When you are finished with the Page Maintenance dialog box choose Close.

Changing Pages

There are several ways to display a different page in a SmartObject file.

To display the next page:

Click on the Next Page button.

To display the previous page:

Click on the Previous Page button:

To jump to a page:

1. Click on the Go To Page button.
2. The Go To Page dialog box appears.
3. In the Page List area click on the page you want to jump to.
4. Choose OK.

   The dialog box is closed and the selected page is displayed.

Working with the Page Color

The SmartObject Editor lets you change the background color of any page. For example, you can make the entire background blue, red, or transparent. If you make the screen transparent when the page is displayed in IconAuthor, the objects on the page will appear to rest directly on top of the existing screen display.

If you display a page with a color background the color replaces any previously displayed information except for any visible, live objects.

Note: While you are viewing a page with a transparent background within the SmartObject Editor, the background appears white. Do not confuse this with a white background which is not transparent.

You can change the screen color to one of 48 basic colors or you can define a custom color.

To change the screen color:

1. Click the right mouse button anywhere on the page background (not on an object).

   The Page menu appears.
2. Choose Color...

   The Color dialog box appears.
3. Choose a color from the color palette and choose OK.

When you accept a color selection the screen appears with the new color. If you create and then show a new page in the same SmartObject file the screen in that new page has the current color.

Managing SmartObject Files

Managing your SmartObject files means understanding basic tasks such as saving your work, opening existing files, and opening new files.

Assign any legal DOS filename to a SmartObject file.   The filename can be one to eight characters with an optional one to three character extension following a period.

Related Topics:

Saving New Files
Saving Existing Files
Saving and Renaming Files
Starting New Files
Opening Existing Files
Printing Files
Path Information

Saving New SmartObject Files

The first time you save a SmartObject file you also name it. If you do not specify a filename extension, the appropriate extension is appended for you. That is, in Page Layout mode the filename ends in .SMT; in Formatted Text mode it ends in .FTT; and in Text mode it ends in .TXT.

To save a new SmartObject file:

1. Choose Save As... from the File menu.

   The Save As dialog box appears.

2. Make sure the dialog box is pointing to the right drive and directory.

   The current path is listed directly below "Directories." Use the Drives drop-down list box to change to another drive and use the Directories list box to change to another directory. Double click on a closed directory folder to open it. When a folder is open any files in that directory (that match the filter in the File Name text box) are listed (greyed) in the list box below the File Name text box.

3. Make sure the Save File as Type box shows the correct kind of file.

   For example, if you are saving a SmartObject file, this box should be set to SmartObject(*.smt). Click on the down arrow to show the available selection of file types.

4. When the dialog box is pointing to the right directory and the file type is correct, you can specify the name you want to assign to the file.

   Choose a filename from the list box below the File Name box or type a new filename into the File Name box.

5. Click on OK.

When you choose OK the dialog box is removed and the file is named and saved. The name of the SmartObject file appears in the title bar.

Saving Existing SmartObject Files

Periodically, it is a good idea to save changes to a SmartObject file that has been saved previously, but now contains unsaved changes.

To save changes to an existing SmartObject file:

- Choose Save from the File menu or press CTRL + S.

  The SmartObject file is saved automatically.

Saving and Renaming SmartObject Files

It is sometimes useful to save a SmartObject file and rename it at the same time. For example, you might want to use DEMO1.SMT as a template. To do so, make a copy of it under a new name, while keeping DEMO1.SMT intact for future use.

If you save and rename DEMO1.SMT with the name DEMO2.SMT, you now have two identical files with different names. Make as many changes as you like to DEMO2.SMT and you will still have a copy of the original DEMO1.SMT.

To save and rename a SmartObject file:

1. Open the file you want to copy.
2. Choose Save As... from the File menu.

   The Save As dialog box appears.
3. Make sure the dialog box is pointing to the right drive and directory.

   The current path is listed directly below "Directories." Use the Drives drop-down list box to change to another drive and use the Directories list box to change to another directory. Double click on a closed directory folder to open it. When a folder is open any files in that directory (that match the filter in the File Name text box) are listed (greyed) in the list box below the File Name text box.
4. Make sure the Save File as Type box shows the correct kind of file.

   For example, if you are saving a SmartObject file, this box should be set to SmartObject(*.smt). Click on the down arrow to show the available selection of file types.
5. When the dialog box is pointing to the right directory and the file type is correct, you can specify the name you want to assign to the file.

   Choose a filename from the list box below the File Name box or type a new filename into the File Name box.
6. Click on OK.

When you choose OK the dialog box is removed and the file is named and saved. The new name of the SmartObject file appears in the title bar.

Starting New SmartObject Files

When you start the SmartObject Editor the work area is blank. You can begin to create the contents of a new file immediately. You can also create a new SmartObject file even if you already have a SmartObject file currently visible in the work area.

To clear the work area for a new SmartObject file:

1. Choose New... from the File menu.

   A new SmartObject file appears *if* the SmartObject file you were working on did not contain unsaved changes.

   If the SmartObject file contained unsaved changes, a message box appears that asks if you want to save the current file. To proceed without saving changes choose No. To save changes before proceeding, choose Yes and a Save As dialog box appears.

2. Make sure the dialog box is pointing to the right drive and directory.

   The current path is listed directly below "Directories." Use the Drives drop-down list box to change to another drive and use the Directories list box to change to another directory. Double click on a closed directory folder to open it. When a folder is open any files in that directory (that match the filter in the File Name text box) are listed (greyed) in the list box below the File Name text box.

3. Make sure the Save File as Type box shows the correct kind of file.

   For example, if you are saving a SmartObject file, this box should be set to SmartObject(*.smt). Click on the down arrow to show the available selection of file types.

4. When the dialog box is pointing to the right directory and the file type is correct, you can specify the name you want to assign to the file.

   Choose a filename from the list box below the File Name box or type a new filename into the File Name box.

5. Click on OK.

When the file has been saved the new, blank file appears.

Opening Existing SmartObject Files

To open a SmartObject file:

1. Choose Open... from the File menu.

   The Open dialog box appears if a) you just started the SmartObject Editor and the drawing area is still empty, or b) you are working on a SmartObject file that does not contain unsaved changes.

   If the SmartObject file contained unsaved changes, a message box appears that asks if you want to save the current file. To proceed without saving changes choose No. To save changes before proceeding, choose Yes and a Save As dialog box appears.

   When you finish using the Save As dialog box the current file is saved and the process of opening an existing file continues.

2. When the Open dialog box appears, make sure the dialog box is looking in the right place for the file you want.

   The current path is listed directly below "Directories." Use the Drives drop-down list box to change to another drive and use the Directories list box to change to another directory. Double click on a closed directory folder to open it. When a folder is open any files in that directory (that match the filter in the File Name text box) are listed in the list box below the File Name text box.

3. Once the dialog box is looking in the right directory you may have to change the value in the List Files of Type field.

   For example to show all files in the current directory, choose All (*.*) in the list box.

4. When the correct file appears in the list box below the File Name text box, click on the filename and choose OK.

   The SmartObject file is opened.

Printing SmartObject Files

SmartObject files can be printed by the page. The information on the page is printed as it appears on the screen.

To print a SmartObject file:

1. Choose Print... from the File menu.

   The Print dialog box appears.

   Note: The Print command uses the current printer selected during Microsoft Windows installation. The printer driver for your printer must be installed on a hard disk drive in your computer system. Use the CONTROL.EXE program in the Microsoft Windows directory to select the current printer and printer communications port configuration.
2. Click to select the page you want to print or choose Select All to select all pages.
3. Choose OK to print the selected page(s), or choose Cancel to close the dialog box without printing.

Path Information

If you are using a customized directory structure to organize the files you create with IconAuthor, the SmartObject Editor, and other editors, you may have to use the Set Path From File... command in the SmartObject Editor File menu.

When you access the SmartObject Editor, by default, it assumes you are working with files located in the paths defined in your master IAUTHOR.PTH file. If you are creating a SmartObject file for an application that uses paths other than those in IAUTHOR.PTH, choose Set Path From File... to select an alternative .PTH file for the SmartObject Editor to use. Once you specify an alternative .PTH file, the SmartObject Editor will know where to find and store the SmartObject and graphics files for that application.

Use the Set Path From File... command as often as necessary. For example, you may open the SmartObject Editor and decide to work on a SmartObject file used by PROJ1.IW. Therefore, you choose Set Path From File... and select PROJ1.PTH. At a later point in time, you decide to work on a SmartObject file used by PROJ2.IW. The files used by PROJ2.IW are stored in a different group of subdirectories. Therefore, you choose Set Path From File... again, and select PROJ2.PTH.

Quitting the SmartObject Editor

When you are finished working with the SmartObject Editor you can exit the program. By default the next time you start up the SmartObject Editor it will remember the settings you were using. For example, it will remember if you were running the editor full screen or if you had all the features such as the grid turned on or off. The SmartObject Editor remembers the state of these features because by default, Save Settings On Exit in the Options menu is toggled on. Choose the option if you want to toggle it off.

To quit the SmartObject Editor:

- Choose Exit from the File menu.

  If you choose exit and have not yet saved the file you are working on, a dialog box appears and asks if you want to save the changes to the file.

To exit without saving changes:

- Choose No.

To exit and save changes:

1. Choose Yes.

   If the file you are saving has been saved (named) on a previous occasion, it is saved automatically.

   If the file has not been saved before (it is "untitled"), the Save As dialog box appears.
2. Enter a filename in the File Name text box.
3. Choose OK.

   The file is saved and the SmartObject Editor is closed.

When you open the SmartObject Editor from a Content Editor, work with the editor, and close it, the name of the file you last worked on is returned to the Content Editor text box from which the editor was started.

AlignHorizontal Property

This property controls the horizontal alignment used for *all* text in a text object.   You can set it to Left, Right, or Center.

Area Property

This property is made up of four numbers separated by commas that describe the area of the object. Example: 100,100,50,50. The first two numbers are the pixel coordinates of the upper left corner of the object. The second two numbers are the width and height of the object (in pixels).

This property can only be manipulated at runtime via the IconAuthor object icons.

BaseLine Property

This property, set to true or false, controls whether an underline appears in a text object to let the user know how many spaces are available for input. For example, if the object has an input style of Currency, BaseLine is true, and InputLimit is set to 10, the underline will extend out to 10 positions to show the user how many digits can be specified. This property is only available for the following input styles: Currency, Financial, Fixed Decimal and Picture.

Bottom Property

This property is the distance (in pixels) from the top of the page to the bottom of the object. Example: 100.

This property can only be manipulated at runtime via the IconAuthor object icons.

Border Property

This property, set to true or false, controls whether a border appears around the graphic.

CharacterCurrency Property

This property controls the character used to indicate a particular unit of currency. The default is the "$" symbol. This property is only used by text objects with an input style of Currency.

CharacterDecimal Property

This property controls the character used to denote a decimal point. The default is the "." symbol. This property is only used by text objects with an input style of Currency, Financial, or Fixed Decimal.

CharacterFalse Property

This property controls the single character a user must enter to indicate a false or negative response if the input style of a text object is Logical. The default is the "F" character. This property is used in conjunction with the CharacterTrue property.

CharacterThousands Property

This property controls the character used to separate every group of three digits to the left of the decimal point.   The default is the "," character.   Example: 1,000,000.00.   This property is only used by text objects with an input style of Currency or Financial.

CharacterTrue Property

This property controls the single character a user must enter to indicate a true or positive response if the input style of a text object is Logical. The default is the "T" character. This property is used in conjunction with the CharacterFalse property.

Checked Property

This property, set to true or false, controls whether a button object that is a check box or radio button style is checked (selected).

CheckedRadioButton Property

This property identifies the specific radio button (in a group of radio buttons) that has been selected by the user. In order to be recognized as belonging to the same group, the radio buttons must all have the same FamilyName setting. You cannot set this property. Rather, you retrieve it at runtime with an ObjGet icon. The ObjGet icon takes the object name of the radio button that is checked and assigns it to this property.

ColorBackground Property

This property controls the background color of listboxes, OLE objects, and most styles of button objects. Example: This property sets the fill color for a button that has a check box style. (If a button has a push button style it does not have this property because its color can't change.) The drop-down list box lets you use the Colors dialog box where you can select a color or create a custom one.

Note: If you make the ColorBackground of an OLE object transparent, the object automatically becomes static. Live objects cannot have transparent backgrounds.

ColorFace Property

This property controls the color used for the face of a text object.   This property can only be manipulated at runtime via the IconAuthor object icons.

ColorFill Property

This property controls the color used for the background of *all* text in a text object.   This property can only be manipulated at runtime via the IconAuthor object icons.

ColorFrame Property

This property controls the color used for the border frame of a text object.

ColorHighlight Property

This property controls the color used for the border highlight of a text object. This property can only be manipulated at runtime via the IconAuthor object icons.

ColorShadow Property

This property controls the color used for the border shadow of the text object. This property can only be manipulated at runtime via the IconAuthor object icons.

ColorText Property

This property controls the text color for listboxes, text objects, and most styles of button objects. Example: This property sets the text color for a button that has a check box style. (If a button has a push button style it does not have this property because its text color can't change.) The drop-down list box lets you use the Solid Colors dialog box where you can select a color or create a custom one.

For text objects, this property can only be manipulated at runtime via the IconAuthor object icons.

CursorName Property

This property controls the way the cursor appears at runtime when it is over the object. The default setting is the arrow-shaped cursor. The drop-down list box lets you use the Cursor Selector to choose one of the available cursors such as Indexed Hand, Down Arrow, and Crosshair.

DecimalPlaces Property

This property controls the number of digits that can appear to the right of the decimal point. The default is 0.

DefaultAction Property

This property lets you set the default action used by an OLE object. The default action occurs when a user double clicks on the object at runtime. Two options for this property are always available: Server Default and None. Server Default means that the action that is automatically designated by the server is used. None means that nothing happens when the user double clicks. Most OLE objects also have other possible actions to choose from. Example: Although the default action for a wave audio file is *play*, you also have the option of setting this property to *edit*.

DeleteProtected Property

This property, set to true or false, indicates whether (at runtime) an object can be deleted by an ObjDelete icon.

DrawStyle Property

This property is available for graphic and OLE objects.

For a graphic, use one of the following:

Scale - The graphic is resized to fit precisely into the available object. The graphic may be stretched or compressed either horizontally or vertically to fit.

Clip - The object contains as much of the graphic as is possible.

Tile - The graphic (in its original size) is repeated in order to fill the area of the object.

Size By Graphic - The object size changes so that it precisely surrounds the entire graphic.

For an OLE object, use one of the following:

Size By Server - The data you embed in the object controls the size of the object.

Size By Object - The size of the object controls how the data appears regardless of how the data appeared in the server.

Editable Property

This property specifies whether a text object is meant to display text or it is meant for a user to input text. The drop-down list options are true and false. The default is false.

Enabled Property

This property, set to true or false, controls whether an object is active, that is, whether a user can interact with it. For example, if you disable a button a user cannot click it. Similarly, if you disable a list box a user cannot scroll it or make a selection from it. When you disable an object any text labels it contains are greyed.

FamilyName Property

This property allows you to optionally specify a family (group) to which an object belongs. If you make an object part of a family, at runtime, you can change a property of multiple objects simultaneously by specifying the change to effect a scope of family.

The FamilyName property provides special functionality for radio buttons (button objects with a radio button Style). When you want a quantity of radio buttons to act as one group (where only one can be selected at any time), make sure to give each radio button the same FamilyName.

FileName Property

This property lets you specify the name of the file that you want to display in a text or graphic object. The drop-down list box lets you use the Directory item to find the name of the file you want to use.

Focus Property

This property allows you to set the focus to a particular object. When an object has focus, it is the object that will respond to any keyboard actions. For example, if a button has focus and the user presses the Return key, that button will be activated. Only one object can have focus at a time. This means that if one object has focus and you give focus to a second object, the first object no longer has focus.

Font Property

This property allows you to set the font used for any text displayed as part of an object. The drop-down list lets you use the Font dialog box to choose a font. Example: You set the font for the text label on a button or the text items in a list box.

Height Property

This property is the height of the object in pixels.   Example: 165.   This property can only be manipulated at runtime via the IconAuthor object icons.

IconFileName Property

This property controls the file used for an icon in two situations. A button object that has an Icon style uses the IconFileName property.   Also, the window object uses this property for the appearance of the icon when the window is minimized.   The drop-down list lets you use the Directory option to select the .ico file you want to use.

InputLimit Property

This property lets you specify the maximum number of characters a user can input into a text object. At runtime, when the limit is reached, the string "MaximumText" is automatically stored in the system variable @_Object_Event.

InputLimitBeep Property

This property, set to true or false, controls whether a system beep occurs when a user has reached the limit for input (as set in the InputLimit property).

InputTerminationRequired Property

This property, set to true or false, controls whether users must press the RETURN key in order for a NotifyOnComplete event to occur after they provide input. If this property is false, the event occurs as soon as the InputLimit property setting is reached.

Example: InputLimit is set to 10, the user enters 10 characters, an event occurs immediately. If this property is true, the user could enter 10 characters but no event occurs until RETURN is pressed.

ItemList Property

This property lets you specify the items that you want to appear in a list box. Separate each item from the next with a semicolon (;). Example: If you specify blue;red;green;black these four items appear in the list.

Label Property

This property lets you set the text that labels a button object.   Example: You set this property to Cancel if you want a Cancel push button

KeyboardTabStop Property

This property, set to true or false, indicates whether a user can use the tab key to highlight this object at runtime.

Left Property

This property is the distance (in pixels) from the left side of the page to the left side of an object. This property can only be manipulated at runtime via the IconAuthor object icons.

LightSource Property

This property controls the imaginary light source that lends shadows to a text object. Set LightSource to one of four directional values: NW, NE, SE, or SW. This property can only be manipulated at runtime via the IconAuthor object icons.

LineSpace Property

This property controls the line spacing in a text object. Set LineSpace to one of three values: 1 (for single spacing), 1.5 (for one and one-half spacing), or 2 (for double spacing). This property can only be manipulated at runtime via the IconAuthor object icons.

Location Property

This property is the pixel coordinates of the upper left corner of the object. Example: 250,125. This property can only be manipulated at runtime via the IconAuthor object icons.

Mask Property

Set this property for text objects that have the Picture input style. The Mask value is a series of special characters that strictly control which characters a user can enter and how many. Use these reserved characters to create a mask:

9 - a number

A- any alphabetical character or space

X - any alphabetical character, number, or space

DD - a two-digit sequence that represents a day (a value from 01-31)

MM - a two-digit sequence that represents a month (a value from 01-12)

YY - a two-digit sequence that represents a year (01-99)

YYYY - a four-digit sequence that represents a year (0001-9999)

B - any character or symbol

! - any character or symbol where any alphabetical character is converted to uppercase

Multiple Lines Property

This property lets you indicate whether a text object allows multiple lines of text to be input. If this property is set to true, text wraps down vertically onto the next line when the cursor reaches the left side of the text objects. If this property is set to false, text does not wrap to the next line. Rather, it scrolls horizontally out of view when the text object becomes full. The user can use the arrow keys to bring text back into view.

NotifyOnClickLeft Property

This property, set to true or false, controls whether an event occurs when a user clicks the left mouse button on an object.   It is available for the text, graphic, button, transparent, and window objects.   When an ObjEvent icon detects that an event has occurred, it places the string "ClickLeft" in the system variable @_Object_Event.   Also, the name of the object is placed in @_Object_Name.   Note that if this property is set to false, a user may be able to click the left mouse button on the object; but an event does not occur.

NotifyOnClickMiddle Property

This property, set to true or false, controls whether an event occurs when a user clicks the middle mouse button on an object. It is available for the text, graphic, button, transparent, and window objects. When an ObjEvent icon detects that an event has occurred, it places the string "ClickMiddle" in the system variable @_Object_Event. Also, the name of the object is placed in @_Object_Name. Note that if this property is set to false, a user may be able to click the middle mouse button on the object; but an event does not occur.

NotifyOnClickRight Property

This property, set to true or false, controls whether an event occurs when a user clicks the right mouse button on an object. It is available for the text, graphic, button, transparent, and window objects. When an ObjEvent icon detects that an event has occurred, it places the string "ClickRight" in the system variable @_Object_Event. Also, the name of the object is placed in @_Object_Name. Note that if this property is set to false, a user may be able to click the right mouse button on the object; but an event does not occur.

NotifyOnComplete Property

This property, set to true or false, is used by OLE and text objects.

OLE objects - Controls whether an event occurs when an OLE object finishes its action.

Text objects (that are editable) - Controls whether an event occurs when a user presses the termination key while editing text in an Editable text object. The default termination key is Enter.

In either situation, when an ObjEvent icon detects that an event has occurred, it places the string "Complete" in the system variable @_Object_Event. Also, the name of the object is placed in @_Object_Name.

NotifyOnInputLimit Property

This property, set to true or false, controls whether an event occurs when a user is entering text in a text object and reaches the input limit. The input limit is set in the InputLimit property. When an ObjEvent icon detects that an event has occurred, it places the string "InputLimit" in the system variable @_Object_Event. Also, the name of the object is placed in @_Object_Name.

NotifyOnSelect Property

This property, set to true or false, controls whether an event occurs when a user double clicks on a list box item or selects an item and presses the Return key. When an ObjEvent icon detects that an event has occurred, it places the string "Select" in the system variable @_Object_Event. Also, the name of the object is placed in @_Object_Name. Note that if this property is set to false, a user may be able to make a selection from a list box; but an event does not occur.

NotifyOnSelectChange Property

This property, set to true or false, controls whether an event occurs when a user changes the selection made in a list box, either by clicking on a different item or pressing an arrow key to move to another item. When an ObjEvent icon detects that an event has occurred, it places the string "SelectChange" in the system variable @_Object_Event. Also, the name of the object is placed in @_Object_Name. Note that if this property is set to false, a user may be able to change the selection; but an event does not occur.

NotifyOnStart Property

This property, set to true or false, controls whether an event occurs when an OLE object starts its action (for example when a wave audio file starts playing). When an ObjEvent icon detects that an event has occurred, it places the string "Start" in the system variable @_Object_Event. Also, the name of the object is placed in @_Object_Name. Note that even if this property is set to false, an OLE object can successfully start its action; but an event does not occur.

ObjectName Property

This property is the unique name that you must assign to the object within the SmartObject editor. At runtime, when you want to change (set) or retrieve (get) the current property of an object, you will need to specify the object name in the ObjSet or ObjGet icon Content Editor.

Also at runtime, when a user acts on an object (for example, clicks on a button) the name of the object that was affected is automatically placed in the system variable @_Object_Name. This information is extremely valuable. Example: You give the user two buttons to click on labeled OK and Cancel. The user clicks on a button. Use a Branches composite made of If icons to check the current contents of @_Object_Name to determine which button was clicked and then branch accordingly.

Rectangle Property

This property is made up of four numbers separated by commas that describe the area of the object. Example: 350,200,75,75. The first two numbers are the pixel coordinates of the upper left corner of the object. The second two numbers are the pixel coordinates of the lower right corner of the object. This property can only be manipulated at runtime via the IconAuthor object icons.

Right Property

This property is the distance (in pixels) from the left side of the page to the right side of an object. This property can only be manipulated at runtime via the IconAuthor object icons.

ScrollBarVertical Property

This property, set to true or false, lets you specify whether the text object has a vertical scroll bar.

SelectedItemData Property

This property identifies the currently selected item in the list. You cannot set this property; it is set by the user at runtime when a list box item is selected.

Example: If the user selects black the property is set to black. Your application can use an ObjGet icon to retrieve this value and store it in a variable. Once in a variable, your application can evaluate the value and take an appropriate action.

SelectedItemNumber Property

This property identifies the number of the currently selected item in the list.

Example: A list box contains three values, slow, medium, and fast. If the user selects medium the property is set to 2 because medium is the second item in the list. Your application can use an ObjGet icon to retrieve this value and store it in a variable. Once in a variable, your application can evaluate the value and take an appropriate action.

SelectionArea Property

This property lets you assign a number to a text, graphic, or OLE object so that it is input selectable. Assign a unique number to each selection area.

ShowPartialItems Property

This property controls whether items in the list box scroll *by partial items* or *by whole items*. If a list box scrolls by partial items, the items appear gradually, similar to the numbers that gradually rotate into view on an odometer. If a list box scrolls by whole items each item pops into view in its entirety

Size Property

This property is the width and height of the object. Example: 200,56. This property can only be manipulated at runtime via the IconAuthor object icons.

Sort Property

This property, set to true or false, controls whether the items in the list box are sorted alphabetically.

State Property

This property lets you set the state of an OLE object to Executing or Idle. If the state of the object is Executing the object's default action occurs. If the state of the object is Idle no action occurs. Example: Your application displays a page with a non-visible OLE object that is defined to play a wave audio sound file. A user cannot double-click on the object to play it because the object cannot be seen. However, an ObjSet icon can set the object's State property to Executing and the sound file plays.

This property can only be manipulated at runtime via the IconAuthor object icons.

Style Property

This property is used by button and timer objects to control the general style of the object. This property can only be manipulated at runtime via the IconAuthor object icons. Within the editor you change the style of the object via the Styles... command in the object's pop-up menu.

For a button object use one of the following:

Push Button
Icon Button
Check Box
Radio Button
Group Box

For a timer object use one of the following:

Count Up
Count Down
Alarm
Periodic

Text Property

This property controls the text that a text object currently contains. For example, if a user types the word blue into an editable text object, the Text property is set to blue. Your application can use an ObjGet icon to take the current Text property value and store it in a variable. This property can only be manipulated at runtime via the IconAuthor object icons.

TextCase Property

This property, set to MixedCase, UpperCase, or LowerCase, controls the case used for alphabetical characters. MixedCase leaves the text in the case in which it was entered. UpperCase converts all text to uppercase. LowerCase converts all text to lowercase.

TextLength Property

This property is set to the number of characters in a text object. This property can only be manipulated at runtime via the IconAuthor object icons.

TimerData Property

This property lets you specify the data used by the timer object. For a Periodic timer, this setting is the amount of time between alarm events. Example: 01:00:30. For a Count Up timer this is 0 (the starting point in time). For a Count Down timer this is the amount of time that passes before the alarm event occurs. Example: 00:00:10. For an Alarm timer this is the specific time of day at which the event should occur. Example: 14:00:00.

Top Property

This property is the distance (in pixels) from the top of the page to the top of the object. Example: 125. This property can only be manipulated at runtime via the IconAuthor object icons.

Visible Property

This value (true or false) indicates whether the object can be seen. When an object is not visible it still exists and can be made visible at some later time.

All object classes that have a Visible property, *except* OLE objects, are also effectively disabled when you make them invisible. Example: If the user cannot see a button or a list box, he or she cannot click or double click on it.

OLE objects are not disabled when they are made invisible. It is true that a user cannot activate the object by clicking or double clicking on it if the object cannot be seen. However, the object can still be activated. OLE objects have a special runtime only property called State. You can use an ObjSet icon to change the State property from idle to executing and the default action associated with the OLE object is executed. Example: An invisible OLE object is defined to play a sound file. An ObjSet icon can set the State property of the object to executing and the sound file plays.

Width Property

This property is the width of the object in pixels. Example: 75. This property can only be manipulated at runtime via the IconAuthor object icons.

WidthEdge Property

This property controls the width of a text object's border edge. Set WidthEdge to the desired number of pixels. This property can only be manipulated at runtime via the IconAuthor object icons.

WidthFrame Property

This property controls the width of a text object's border frame. Set WidthFrame to the desired number of pixels. This property can only be manipulated at runtime via the IconAuthor object icons.

SmartObject Editor Contents

The Contents lists the Help topics available for the SmartObject Editor.   Use the scroll bar to see entries not currently visible.

Keyboard

SmartObject Editor Keys

Commands

Edit Menu
File Menu
Help Menu
Objects Menu
Options Menu
Page Menu

Procedures

Changing Fonts
Changing Text Colors
Creating Input Selectable Objects
Creating Objects
Displaying Video in the SmartObject Editor
Editing Text
Entering Text
Live Object Maintenance
Making a Text Object Editable
Making an Object Live or Static
Managing Files
Managing Pages
Objects
Object Alignment
Paragraph Alignment and Spacing
Path Information
Quitting the SmartObject Editor
Selecting and Editing Objects
Selecting Text
Setting Object Properties
Typing IconAuthor Variables
Using ASCII Text Mode
Using Formatted Text Mode
Working with Text in a Text Object
Working with the Style of a Text Object

SmartObject Editor Keys

The following accelerator keys and key combinations are available in the SmartObject Editor:

Menu	Command	Key(s)
File	Save	Ctrl + S
Edit	Cut	Shift + Del
	Copy	Ctrl + Ins
	Paste	Shift + Ins
	Clear	Del
Page	New Page...	Ctrl + W
Object (Order)	Bring To Front	Ctrl + U
	Send To Back	Ctrl + D
Options	Full Screen	Ctrl + F
	Hide Tools	Ctrl + T

Additionally, you can use keys to perform text editing functions within a text object.

Key	Function
arrow keys	The up, down, left, and right arrow keys move the insertion point to a new location.
Home	Moves the insertion point to the beginning of the current line.
End	Moves the insertion point to the end of the current line.
Page Up	Scrolls up in the text block.
Page Down	Scrolls down in the text block.
Delete	Deletes the character to the right of the insertion point.
BackSpace	Deletes the character to the left of the insertion point.

SmartObject Editor Commands

To get help with a command, choose the appropriate menu.

File Menu

New...
Open...
Save
Save As...
Delete
Page Setup...
Print...
Printer Setup...
Set Path From File...
Exit

Edit Menu

Cut
Copy
Paste
Clear
Copy To...
Paste From...

Page Menu

- New Page...
- Page Maintenance
- Page Properties
    - Color...
    - Page Name...
- Next
- Previous
- Go To...

Objects Menu

- Tools
    - Select Object
    - Button
    - Graphic
    - Listbox
    - OLE
    - Text
    - Timer
    - Transparent
- Object Properties (variable menu items)
    - Properties...
    - Object Type...
    - Button Styles...
    - Item list...
    - Insert New Object...
    - Convert Data To Static...
    - Paste
    - Paste Links
    - Links
    - Block Styles...
    - Input Styles...
    - Timer Styles...
    - Fonts...
    - Text Color...
    - Fill Color...
    - Text Styles...
    - Left
    - Center
    - Right
    - Single
    - 1 1/2
    - Double
- Alignment
    - Left
    - Right
    - Top
    - Bottom
    - Center Vertical
    - Center Horizontal
- Order
    - Bring Forward
    - Send Backward
    - Bring To Front
    - Send To Back

Options Menu

Full Screen
Hide Menu
Grid Alignment...
Confirm Clear
Overlay Mode
Ribbon Bar
Status Bar
Save Settings On Exit

Help Menu

Index
Keyboard
Commands
Procedures
Using Help
About the SmartObject Editor...

SmartObject Editor Procedures

Objects

Button Objects
Graphic Objects
Listbox Objects
OLE Objects
Text Objects
Timer Objects
Transparent Objects

Working with Objects

Creating Objects
Selecting and Editing Objects
Setting Object Properties
Making an Object Live or Static
Live Object Maintenance
Creating Input Selectable Objects

Putting Text in Text Objects

Entering Text
Scrolling Text
Typing IconAuthor Variables
Editing Text
Changing Paragraph Alignment and Spacing
Changing Fonts
Changing Text Colors
Working with Text Styles
Working with the Block Style of a Text Object

Working with Pages

Creating New Pages
Renaming Pages
Copying Pages
Deleting Pages
Importing Pages
Changing Pages
Page Color

Working with Files

Saving New Files
Saving Existing Files
Starting New Files
Opening Existing Files
Printing Files
Path Information

Miscellaneous

Quitting the SmartObject Editor
Displaying Video in the SmartObject Editor