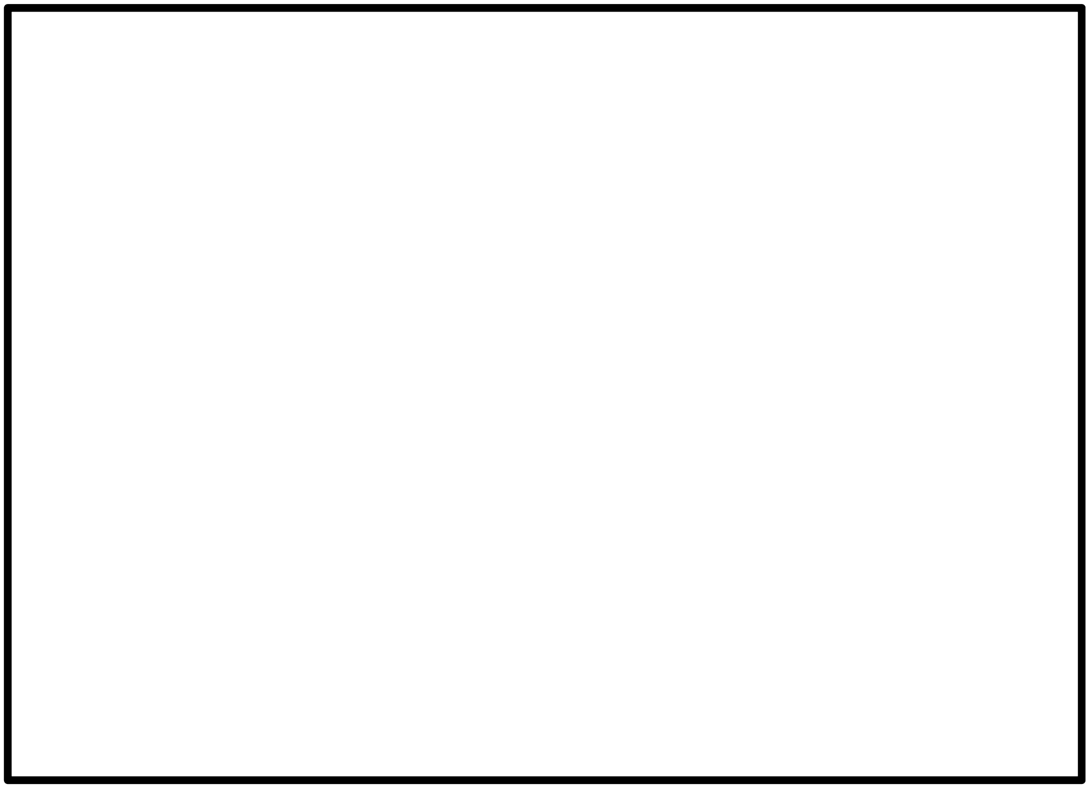
Everest Authoring System

Tutorial

Doc-To-Help Standard Manual

By Intersystem Concepts, Inc.



Acknowledgements

We would like to thank the following people for their help during the development of this version:

Malcolm Burk
Gary Klimple
John Simpson
Don White

Trademarks

For More Information & Technical Support

Contact your vendor, or the developer:

Intersystem Concepts, Inc.
P.O. Box 477
Fulton, MD   20759
USA

410-531-9000
301-854-9426 (fax)

http://www.insystem.com

1 2 3 4 5 6 7 8 9 0

Table of Contents

Installation

What You Need

Requirements

To successfully install and author with Everest, you need the following:

- a computer with an 80386 microprocessor (or better)
- 4M of memory (8M or more suggested for authoring)
- 10M of available hard disk space
- VGA display capability (SVGA suggested for authoring)
- a mouse
- Microsoft Windows 3.1 or 95 (or compatible)

License Agreement

If you need an official copy of the Everest license agreement on paper, please contact us to request one. An electronic copy can be found in the file LICENSE.TXT supplied with Everest.

Please review the license agreement before starting; installing and/or using Everest indicates your acceptance of the agreement.

Installing

To Install Everest

If you have not already installed the Everest software on your computer, here are the steps for doing so.

1. Place Disk 1 in floppy disk drive A:.
2. From the Windows Program Manager File menu, select Run.
3. When the dialog box appears, type

   ```
   a:setup
   ```
4. Follow the prompts of the setup program.
5. Note that some versions of Everest require additional installation steps. You will be notified of such additional steps (if any) via instructions on the screen.
6. That completes the installation. To run Everest and begin authoring, double click on the Author icon.

Also Check README.TXT

For the latest information about Everest, examine the README.TXT file (if any) located in the directory in which you installed Everest. README.TXT is an ASCII text file that can be viewed with the Windows Notepad or the DOS EDIT program.

Everest Quick Start

Using the Quick Start

What is the Quick Start?

The Quick Start chapter will introduce you to the Everest Authoring System in less than an hour. Because it is designed for speed, the Quick Start does not go into much detail. You will find more information and more practical examples in the other chapters of the Tutorial.

What is Everest?

Everest is a group of software programs that help authors create interactive multimedia applications. Everest is especially well suited for creating Computer-Based Training projects.

Everest employs a page and book metaphor. Via the AUTHOR program, you create pages visually, and arrange them within books. One or more books are known as a "project". To check for proper operation, you can test run your projects directly within the AUTHOR program.

End users of your project "read" the book(s) with help from Everest's ERUN player program. If you wish, you can regulate user access to your projects and generate usage reports via Everest's INSTRUCT administrators program. The ERUN and INSTRUCT programs are not supplied with the Everest Free Authoring Software; they can be obtained by purchasing the necessary licenses.

Prerequisites

As prerequisites, you should be familiar with Microsoft Windows, and with at least one Windows-based application.

Objectives

After completing the Quick Start you will be able to:

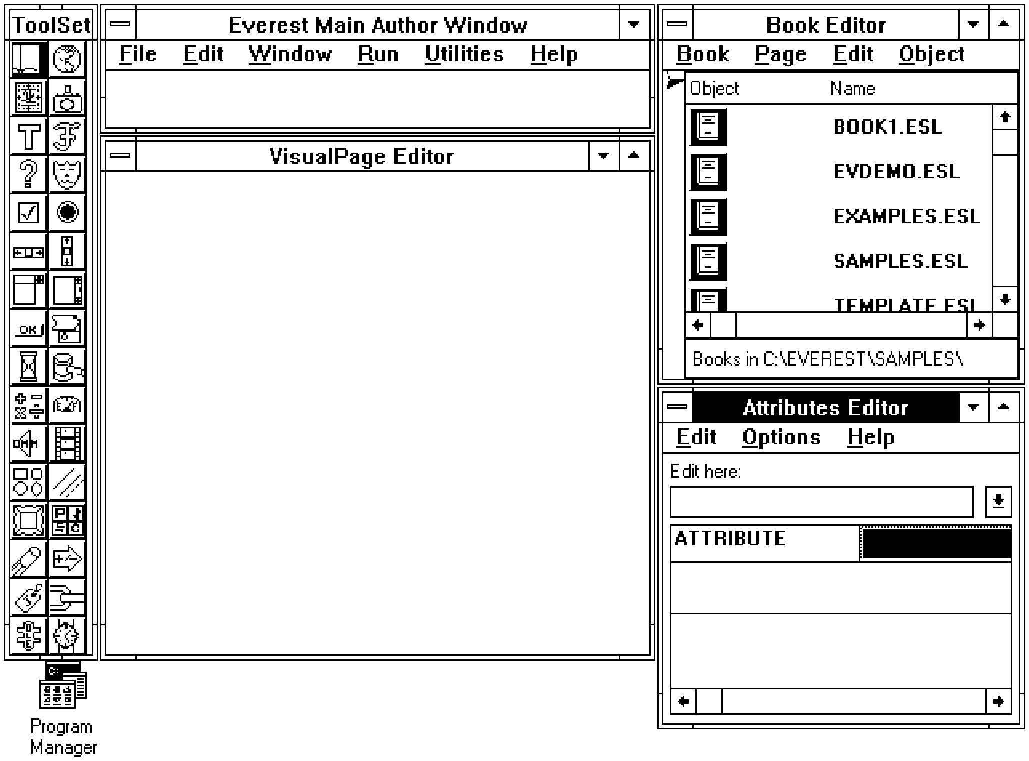
- build pages by dragging icons from the ToolSet
- create and edit text
- change text size, location, color and font
- edit object attributes
- build simple user interaction
- display graphics
- overlay graphics
- employ special visual effects
- use object instancing
- connect pages via branching instructions
- test run your project with Everest's debugger
- edit a page "on-the-fly" while debugging

Time to Complete

An experienced Everest author can create the pages in this exercise in about 3 minutes.   Since you will be using the software for the first time, please allow yourself more time to peruse, experiment and learn.

Introducing the Editing Windows

If you have not already done so, install Everest on your computer.   After the software has been installed, double-click on the Author icon to start the program. After a few seconds, several windows will appear on your monitor, as shown below.



BOOK1.ESL
EVDEMO.ESL
EXAMPLES.ESL
SAMPLES.ESL
TEMPLATE.ESL

Books in C:\EVEREST\SAMPLES\

The Open Windows

Here is a brief description of each of the open windows:

Main Author Window

The window at the top middle is known as the Main Author Window, or simply the Author window. From this window you access the major Everest features, such as opening and saving files, printing, etc. If this window becomes hidden behind others, you can bring it to the front by pressing Ctrl+M.

ToolSet Window

The window along the left edge of the monitor is the ToolSet window. It contains icons for a Textbox, Picture, Multimedia, etc. You build pages by dragging icons from the ToolSet and dropping them on either the VisualPage Editor or the Book Editor.

VisualPage Editor

The VisualPage Editor occupies the center of the monitor. In this window you view the appearance of a page as you create it (What-You-See-Is-What-You-Get).

Book Editor

The Book Editor appears near the upper right corner. The Book Editor lists books (.ESL files). When a book is open, the Book Editor also shows the pages in that book. The Book Editor serves as a flowchart of your project. The icons currently shown are for books that were included with Everest's sample files.

Attributes Editor

The Attributes Editor (also called the Attributes window) on the right displays the properties of the icons (objects) you put on the pages of your books. For example, ForeColor is an attribute of Textboxes that controls the foreground color of the text displayed within.

Resizing the Windows

When you start the Everest authoring software for the first time, it squeezes its editing windows into a 640 x 480 pixel display area, the typical minimum Windows screen resolution, in order to display all the important ones initially. If you are running Microsoft Windows at a higher resolution, these editing windows will appear cramped in the upper-left corner of your monitor.

Recommendations for 640 x 480

However, if you are running Windows right now in 640 x 480 resolution, Everest’s editing windows will occupy most of your monitor. If possible, for greatest ease, we recommend you run Windows at an 800 x 600 (or higher) resolution while authoring. As part of this Tutorial (and probably part of your own projects) you will be constructing pages of a 640 x 480 size. If you author at a higher resolution than this, you’ll be able to see your entire page and the editing windows at the same time.

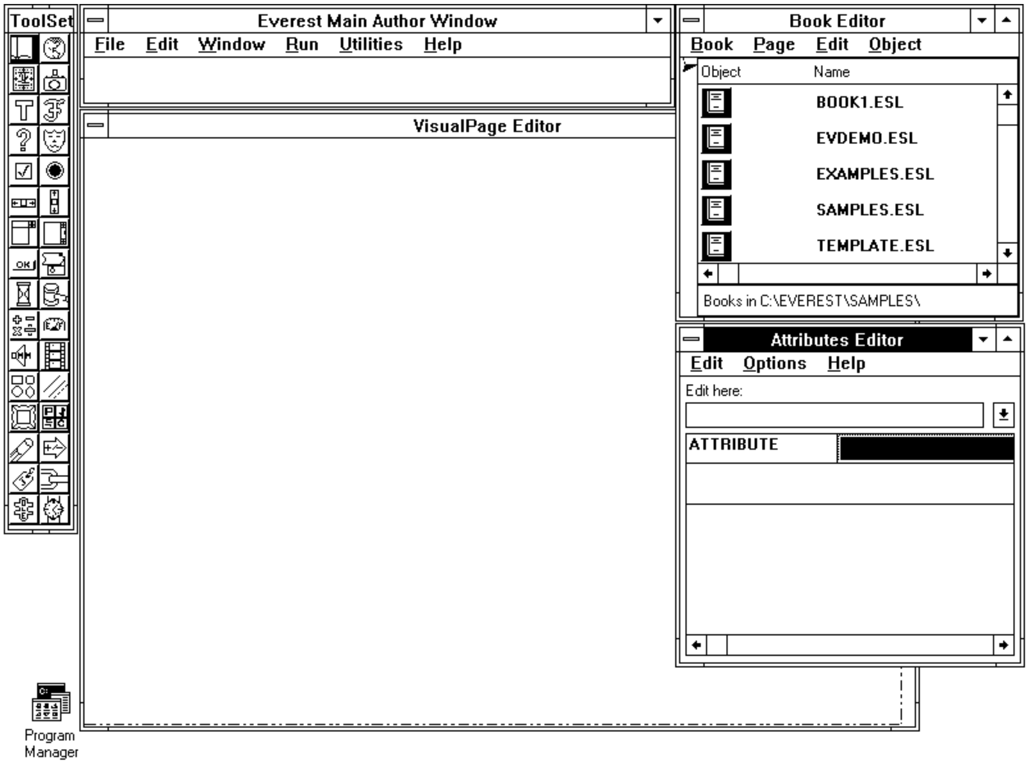
If you are limited to authoring in a 640 x 480 resolution, you can try employing an Everest feature so new that, as of this writing, it is still considered experimental. That new feature is a scrollable VisualPage editor.   To toggle it on and off, either press Ctrl+Alt+S, or choose the feature from the main Author window's Edit menu. This feature lets you scroll the VisualPage editor while authoring...yes, it's a bit tedious, but much better than only being able to see the upper-left corner of the page you are authoring!

Recommendations for 800 x 600

In 800 x 600 resolution, you can see the entire 640 x 480 sized page you are creating, and still have room for Everest's editing windows.

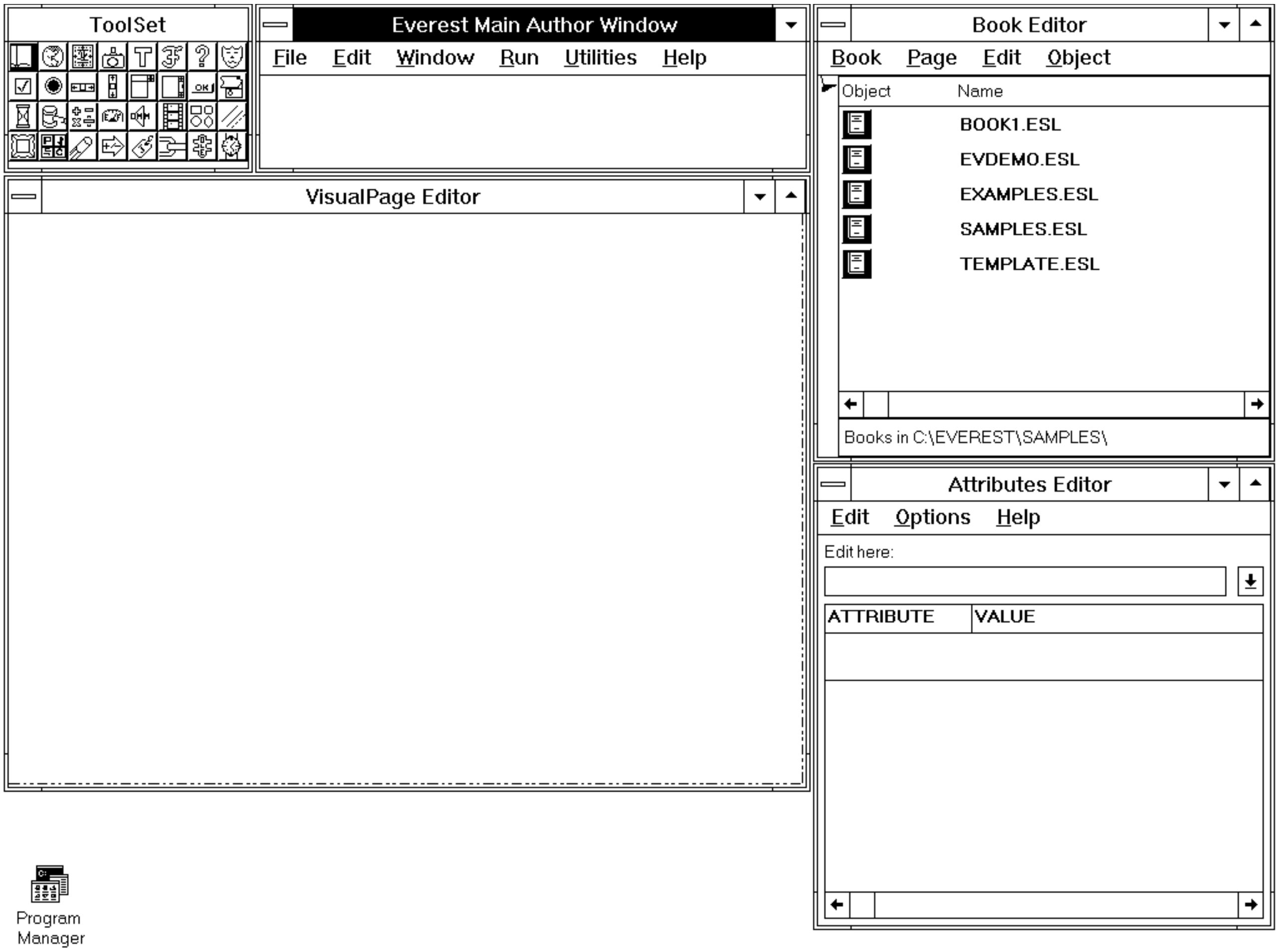
Most authors make the VisualPage editor just large enough to see the dashed lines within.   By default, these dashed lines delineate a 640 x 480 area: the typical minimum display size for most end user's computers.   As you create a page, if you place its contents within this area, the user will not need to scroll the page to view it.

If you are running Windows in a 800 x 600 resolution, we recommend an arrangement like that shown below (notice how the VisualPage editor is just large enough to make the dashed lines visible, but is overlapped by the Book Editor and Attributes Editor):



Recommendations for Higher Resolutions

If you are running Windows in a 1024 x 768 (or higher), you can have a 640 x 480 sized VisualPage editor, without overlap from the other editing windows.   We recommend something similar to the following:



Object	Name
	BOOK1.ESL
	EVDEMO.ESL
	EXAMPLES.ESL
	SAMPLES.ESL
	TEMPLATE.ESL

Books in C:\EVEREST\SAMPLES\

ATTRIBUTE	VALUE

If you prefer a toolbar appearance, make the ToolSet window very wide.

At this point, feel free to resize and relocate the windows as you see fit. You can relocate a window by dragging it by its caption, or change its size by dragging its border. Everest can remember your window arrangement in its EVEREST.INI file.

Creating the First Page

When you create a project with Everest, you create books that contain pages that contain icons (objects). This hierarchy resembles:

```
Books
        Pages
                Icons (Objects)
```

The Book Editor displays your books, pages and objects arranged in a similar indentation. So, let's get started. The first thing to do is create a book to hold the first page. But, wait, where should that book be stored on the computer?

Getting Started - Where Do Things Go?

When you start the AUTHOR program for the first time, it assumes you want to save your work in the directory that contains Everest's samples. That's a good location to use for now, because this Quick Start and Tutorial will employ various sample files (mostly graphics files) stored there.

To verify you are working in the samples directory, look at status line at the bottom of the Book Editor. It should read something like "Books in C:\EVEREST\SAMPLES\". If you installed Everest in something other than the C:\EVEREST directory, then this directory should be listed, followed by \SAMPLES\.

Help! I'm Not in Samples!

If the directory is correct on your computer, you can skip this step. However, if you are not currently working in the EVEREST\SAMPLES\ directory (perhaps because a previous user of Everest has changed to another location), you should now change it back. Here's how: in the main Author window

from the File pull-down menu, choose New

the "Enter Name of New Page" window will open in the column on the left, locate the \EVEREST\SAMPLES subdirectory

after you find the SAMPLES directory, double click on it to open it

click on the OK button

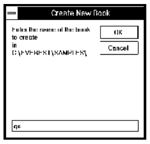
New Book

To hold the pages you'll create during this Quick Start, you should create a new book. To do so, in the Book Editor window

from the Book pull-down menu, choose New

Everest will prompt you to enter the name of the new book. Enter

```
qs
```

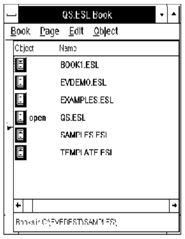


Enter the name of the book to create in C:\EVEREST\SAMPLES\

qs

In the event a book named QS already exists, Everest will alert you; try again, this time append your initials after QS. Book names can consist of up to eight characters.

After you have done this, an icon for the QS.ESL book should appear in the Book Editor. QS.ESL is the name of the disk file Everest created to store the contents of the book. Everest will automatically open the (as yet, empty) book for you.



Object	Name
	BOOK1.ESL
	EVDEMO.ESL
	EXAMPLES.ESL
open	QS.ESL
	SAMPLES.ESL
	TEMPLATE.ESL

Books in C:\EVEREST\SAMPLES\

New Page

As shown above, the word "open" should appear in the Book Editor within the QS.ESL line to let you know that book is open and ready for editing. If (and only if) "open" is *not* displayed, double click on the QS.ESL line in the Book Editor.

You are now ready to create a new page in this book. So, in the Book Editor window

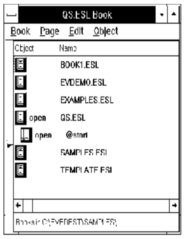
from the Page pull-down menu, choose New

The @ symbol is Shift+2 on most keyboards.

Everest will prompt you to enter the name of the new page. The name of the first page in any book should always be @start, so enter

```
@start
```

After you do so, the name @start should be displayed within the caption of the VisualPage editor. Additionally, your Book Editor should now have a page icon, and resemble:



Object	Name
	BOOK1.ESL
	EVDEMO.ESL
	EXAMPLES.ESL
open	QS.ESL
open	@start
	SAMPLES.ESL
	TEMPLATE.ESL

Books in C:\EVEREST\SAMPLES\

A Quick Look in the ToolSet

The VisualPage window is empty and ready for you to put objects in it.   Objects are things such a Buttons, Textboxes, Pictures, etc.   In Everest, each object is represented by an icon.   The icons from which you can choose are displayed in the ToolSet window.

The ToolSet window resembles (yours may look different: the icons shift to fill the window):

What's That Icon?

We've always found it frustrating to see an icon in some software package and have no idea what it represents.   So, we designed Everest to reveal the icon names easily. Try it now: simply slide your mouse over the icons in the ToolSet.   As you do so, the name of each is displayed as the caption of the ToolSet window.

The Layout Icon

As you were moving your mouse over the ToolSet icons, you may have seen one named "Layout". The Layout icon is an important one. When the end user views the pages of your project, they appear within a window. The Layout controls the appearance of this window, including such attributes as size, caption and background.

The Layout icon is the globe.

Let's put a Layout into your @start page. To do so, drag a Layout icon from the ToolSet and drop it on the VisualPage Editor. In case you are not familiar with drag and drop, here are the steps:

> Choose: click the left mouse button mouse on the Layout icon, and continue to hold down the button as you slide the mouse over to the VisualPage Editor.
>
> Drag: you will observe a square box gliding across the screen as you slide the mouse.
>
> Drop: when the pointer is anywhere inside the VisualPage Editor, release the mouse button. That's it!

Nothing Appears?

After you drop the Layout icon, nothing changes in the VisualPage Editor. That's as it should be.

If the Book Editor is hidden, press Ctrl+B to bring it to the top.

However, look in the Book Editor. The Layout icon now appears there indented beneath the icon for the @start page. It should have the name @start_layout_A. This is the default name Everest assigns (you can change it if you want; for now leave it as is).

If the Attributes window is hidden, press Ctrl+A to bring it to the top.

Also, look in the Attributes window. It should now list attributes of the Layout such as Condition, SaveAsObject, Left, Top, Width, Height, etc. These attributes begin with default values.

640x480 is a common Windows page resolution.

Until you have learned more about Everest, it is best to leave the Layout object's attributes set to their default values. This will produce a window that is 640 x 480 pixels in size.

Add a Textbox

It's time to put some text in this page. So, from the ToolSet

> drag a Textbox object (the T icon), and drop it anywhere inside the VisualPage editor.

When you drop an icon in the VisualPage editor, Everest adds it to list in the Book Editor. The icon will be inserted at the location marked by the small pointer along the left edge of the Book Editor window. Everest automatically advances this pointer as you add more icons. For special purposes, you can move the pointer manually by dragging it with the mouse. For now, leave it as is (now it should be positioned just after the Textbox icon).

Something Appears!

This time, something appears within the VisualPage editor. You should see a rectangular box with eight very small boxes around its edge. This is the Textbox with eight sizing handles.

Entering Text

The focus is often called the "highlight."

To enter text in your new Textbox object, first

click on the Textbox to focus on it.

A small cursor will appear inside the box to let you know you can type. Type the following:

```
This is my first Everest page.
```

Everest automatically word-wraps the text as you type it.

Focus for Editing

By the way, now that your page has two icons (Layout and Textbox), how does Everest know which to show in the Attributes window? Everest uses the object that has the focus. There are two ways you can tell Everest to focus on an object: 1) click on the object in the VisualPage editor, 2) click on the object's icon in the Book Editor.

A single click on an icon in the Book Editor simply loads that object's attributes into the Attributes window. A double click on an icon also brings the Attributes window to the front...a handy feature if your authoring windows overlap.

For now, you should keep the focus on the Textbox.

Resizing and Moving the Textbox

This is a good time to try changing the size of your Textbox. Simply point with the mouse at one of its eight sizing handles (the small boxes near the corners and sides), then press and hold down the LEFT side mouse button, and drag the mouse. When you release the mouse button, Everest resizes the Textbox.

Or: hold down the Shift key and press the left button.

To move the whole Textbox, again point at one of the sizing handles, but this time press and hold the RIGHT side mouse button, and drag.

Saving the Page

And that's it (for the moment). Now you can save this (very simple) page to disk. To do so, in the Book Editor

from the Page pull-down menu, choose Save

Everest saves pages so quickly, you might not even realize it has done so; however, if Everest has a problem saving a page, it will let you know.

What Are Those Asterisks?

If you are very observant, you may have noticed that there had been asterisks (*) displayed near the Layout and Textbox icons in the Book Editor. When you saved the page, the asterisks disappeared.

What are they? It's simple: the Book Editor displays an asterisk next to the icon of any object that you change. Creating new objects counts as changing. When you saved those changes, the asterisks were removed.

Don't stop now! The next topic shows how to make your Textbox object look spiffier with colors and fonts.

Changing Colors and Fonts

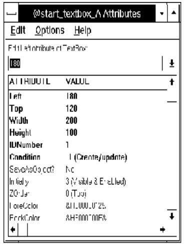
Objects, such as Textboxes, that you add to a page start with certain colors and fonts, specifically, those that you have configured in your copy of Windows. This topic tells you how to improve the Textbox's appearance through the use of color and font changes.

The Attributes Window

Colors and fonts can be modified via the Attributes window. The Attributes window lists the properties of the current object, which, in this case, should be the Textbox.

Right now, the caption at the top of the Attributes window should read "@start_textbox_A Attributes." If you can't find the Attributes window, it may not be open, or perhaps is hidden behind other windows. If necessary, you can bring it into view by pressing Ctrl+A.

If the window does not show the attributes for the Textbox, click on the Textbox in either the VisualPage Editor or the Book Editor.



Edit Left attribute of TextBox

180

ATTRIBUTE	VALUE
Left	180
Top	120
Width	200
Height	100
IDNumber	1
Condition	1 (Create/update)
SaveAsObject?	No
Initially	3 (Visible & Enabled)
ZOrder	0 (Top)
ForeColor	[illegible]
BackColor	[illegible]

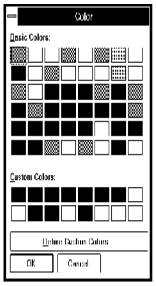
Changing the Foreground Text Color

Tip: to quickly locate an attribute, press Ctrl + Shift+ first letter of the attribute's name. Repeat if needed.

The Attributes window displays properties of the current object, in this case, the Textbox. A two-column grid lists the names of the attributes, plus the value of each. Find the attribute named "ForeColor" (you might need to scroll the contents via the vertical scroll bar inside the Attributes window) and

double click on ForeColor

Doing so opens the standard Windows Color dialog box:



The foreground color is the color of the text (the letters).

With your mouse,

click on a color you want as the foreground,

then click on the OK button in the Color dialog box.

Everest instantly replots the Textbox with the color you chose.

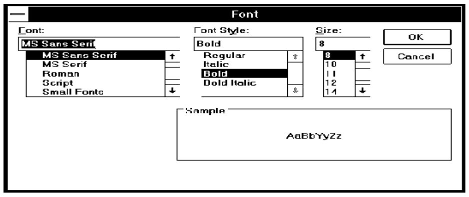
Changing the Font

Now, look in the Attributes window and

find the attribute named "FontName"

double click on FontName

to open the standard Windows Font dialog box:



Font: MS Sans Serif

Font Style: Bold

Size: 8

Here you can choose a font, style and size (go ahead and try whatever you like). The Font dialog box displays some sample characters so you can instantly view the appearance of your selections. When you click on OK, Everest applies your choices to the Textbox.

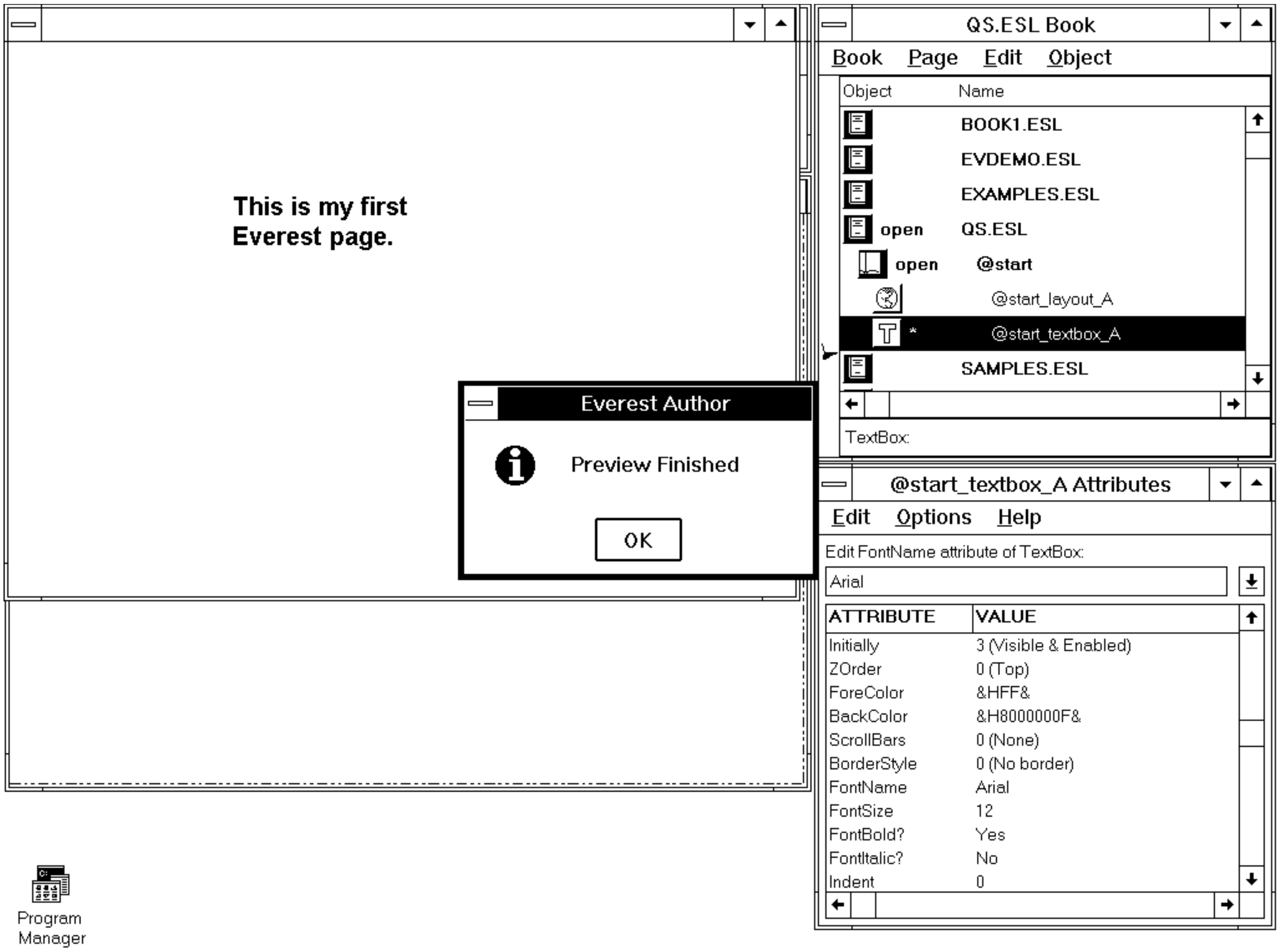
Depending on the font you selected, and its size, you might wish to resize the Textbox. Use the same technique you did in the last topic (remember: left mouse button resizes, right mouse button moves).

Instant Preview

Everest's instant preview feature is a handy one. It lets you run the page that you are currently editing so you can view it as a user will. Try it now; in the Author window

from the Run pull-down menu, choose Preview.

After doing so, your display should resemble:



This is my first
Everest page.

Preview Finished

Object	Name
	BOOK1.ESL
	EVDEMO.ESL
	EXAMPLES.ESL
open	QS.ESL
open	@start
	@start_layout_A
*	@start_textbox_A
	SAMPLES.ESL

TextBox:

Edit FontName attribute of TextBox:

Arial

ATTRIBUTE	VALUE
Initially	3 (Visible & Enabled)
ZOrder	0 (Top)
ForeColor	&HFF&
BackColor	&H8000000F&
ScrollBars	0 (None)
BorderStyle	0 (No border)
FontName	Arial
FontSize	12
FontBold?	Yes
FontItalic?	No
Indent	0

You noticed that Everest displayed your page with the Textbox, and immediately followed that with a "Preview Finished" message box. At this point, your page is very simple, so running a Preview is not very dramatic. In fact, it looks virtually identical to the VisualPage editor. The benefits of Everest's instant preview will become more apparent once your page includes multimedia, animation and special effects.

If the page had contained user interaction and a Wait icon, instead of showing the "Preview Finished" message box, Everest would pause for a response.

Saving Changes

Before moving to the next topic, why not save your changes? If the Preview Finished message box is still displayed, click on its OK button. Then, in the Book Editor window

from the Page menu, choose Save.

Background Images

So far, the @start page is rather plain. Let's make it more visually attractive, starting with the background.

As described earlier, the appearance of the window is controlled by the Layout icon. The Layout determines the graphics displayed in the background of the window. The background is the area of the VisualPage Editor behind all other objects, such as Textboxes.

Editing the Layout

To change the attributes of the Layout icon, you must first make it the current object. To do so,

click on the Layout icon (globe icon) *in the Book Editor*.

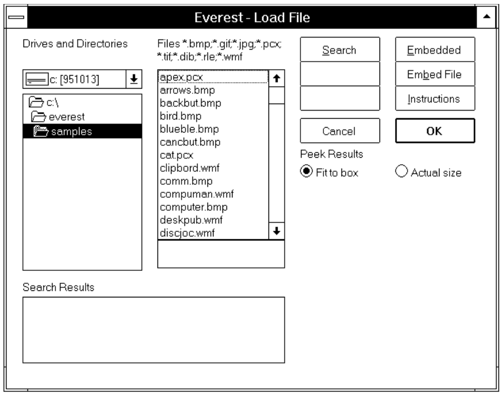
Doing so loads the Layout's attributes into the Attributes window.

Scroll down the list of attributes until you find the one named BgndPicture.

Then

double click on BgndPicture

to open Everest's Load File dialog window, which resembles:



c: [951013]

c:\
everest
samples

apex.pcx
arrows.bmp
backbut.bmp
bird.bmp
blueble.bmp
cancbut.bmp
cat.pcx
clipbord.wmf
comm.bmp
compuman.wmf
computer.bmp
deskpub.wmf
discjoc.wmf

Peek Results
● Fit to box ○ Actual size

The Load File Dialog Window

The Load File window helps you select various external files to link to or embed in your book. We want the sample file STARS.GIF, so

scroll down to stars.gif

and

click once on stars.gif to highlight it

Before we use the file, let's peek at it. With STARS.GIF highlighted

click the Peek button

Peek is very handy when you want to know what the file contains.

With STARS.GIF still highlighted,

click the OK button

to tell Everest this is the file you want to link to.

Tiling

After the Load File window closes, the STARS.GIF image should appear inside the VisualPage window. STARS.GIF is a small image, so it will occupy only the top left corner of the VisualPage window. To fill the window with the image, you want to "tile" it.

First make sure the Attributes window is still showing the attributes for the Layout object (the caption should read "@start_layout_A Attributes"). If it does not, in the Book Editor, double click on the Layout icon.

Then, find the attribute named "Tile?". An attribute that has a ? at the end of its name is called a "Boolean" attribute, or more simply, a "Yes/No" attribute. The value of such attributes can be only Yes or No.

By default, Tile? is No. So,

> double click on the Tile? attribute

to change it to Yes.

When you do so, the STARS.GIF image should replicate to fill the entire VisualPage window.

Overlaid Textbox

Try running a Preview again; in the Author window

> from the Run pull-down menu, choose Preview.

Notice how the Textbox obliterates a rectangular area of the STARS.GIF background. By default, Textboxes are not transparent. Sometimes this is desirable, other times it is not. For now, leave it as is. On the next page we'll create a transparent Textbox.

Save

It's good to develop a habit of saving your work periodically, so

> click OK in the Preview Finished window.

In the Book Editor,

> from the Page menu, choose Save

to save the changes.

Adding a Next Page Button

So far, the @start page is rather simple. In most cases, you'd want something more to act as a title page for your book. For example, you might overlay a picture. However, rather than do that on the @start page itself, we'll instead add a button to it that branches to a second page. The second page will add a picture to the window. This exercise will help you learn about branching.

Adding a Button

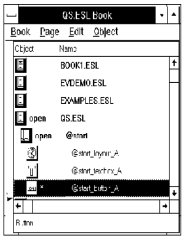
To add a Button object to your @start page, from the ToolSet

drag the Button icon (the one with OK on it)

and drop it on the VisualPage editor. When you drop it, notice that Everest adds the Button to the VisualPage editor, as well as to the Book Editor.

The Book Editor

Your Book Editor should now resemble:



Object	Name
	BOOK1.ESL
	EVDEMO.ESL
	EXAMPLES.ESL
open	QS.ESL
open	@start
	@start_layout_A
	@start_textbox_A
*	@start_button_A

The Book Editor is basically a vertical flowchart of the pages that make up your book, as well as the icons that make up each page. The order of the icons in the Book Editor, from top to bottom, expresses the order in which they are displayed to the user at run time.

Creating the Button Caption

Now, add a caption to the Button. To do so, first focus on the Button object

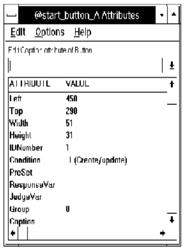
click on the Button in either the VisualPage editor or Book Editor.

Double clicking on Caption opens the Character Table window; close it via its close box or by pressing Alt+F4.

Then, in the Attributes window

click on the Caption attribute.

Your Attributes window should now resemble:



Edit Caption attribute of Button

ATTRIBUTE	VALUE
Left	450
Top	290
Width	51
Height	31
IDNumber	1
Condition	1 (Create/update)
PreSet	
ResponseVar	
JudgeVar	
Group	0
Caption	

Above the list of attributes the small edit field should now be labeled "Edit Caption attribute of Button." In this field, type the word

```
Next
```

and

press Enter.

The word Next should appear on your Button (if it does not, try changing the CaptionColor attribute).

Changing the Button's Name

The default name Everest assigned the Button is something like @start_button_A. As an exercise, you should change this to make the Button easier to reference later. To do so, in the Attributes window

from the Edit menu choose Rename.

A small input window will appear. Change the default name to

```
nextbutton
```

Avoid putting blank spaces within a Name. When done typing, press Enter. The new name should be displayed in the Book Editor next to the Button icon.

SaveAsObject?

Back in the Attributes window,

find the attribute named SaveAsObject

set the SaveAsObject attribute to `Yes` (recall you can double click).

What does SaveAsObject do? When set to Yes, it tells Everest to save a "master" copy of the object independently of the page. Objects saved in this manner can be reused easily on other pages. Plus, if you later need to change something about the object (for example, the color), when you change the master, Everest automatically updates all the pages that employ that object.

Some Windows-based authoring systems hog 50K bytes per page; that's 2500% more than Everest!

SaveAsObject not only makes your project easier to maintain, but since it creates reusable objects, also minimizes the total amount of disk space the project occupies. The average Everest page needs only 2K bytes of disk space for storage. The smaller, the better, especially if you will be distributing your projects electronically, such as via the Internet.

Other Attributes

If you feel adventurous, also change the color, font and/or size of the Button. Notice how the Attributes window displays the location and size of the Button (Left, Top, Width and Height) in units of pixels. You can edit these attributes directly by typing in new numbers, but most authors prefer using the sizing handles in the VisualPage Editor.

Specifying the ClickEvent

This is an important step. At the end of the list of Button attributes is one named ClickEvent. This is where you tell Everest what to do when the user clicks the Button. For example, you might tell Everest to start playing audio, or perhaps download some HTML text from the Internet, or calculate the user's score on an exam, or just about anything.

However, there is one thing you cannot do *directly* from the ClickEvent: you cannot branch to a different page. Instead, Everest requires that your branching instructions appear within Wait objects (which will be discussed more in a moment). This approach gathers all the branching paths together into one object, and helps prevent so called "spaghetti pages" which branch willy-nilly and are very difficult to debug.

So, How Is It Done?

While you cannot branch *directly* from the ClickEvent, you can do so *indirectly*. Here's how (don't worry, it's much easier than it sounds): in the ClickEvent you specify what is called an "event code". An "event code" is nothing more than a message. You get to decide what this message will be...it can be a number, a word, a phrase, anything.

For example, you could set ClickEvent to the word "next". At run time, when the user clicks the Button, the event code (message) "next" will be generated. If you have a vivid imagination, pretend the Button is an impatient director watching auditions for a play. When the director has seen enough, he yells "NEXT!" That's what this Button will do when the user clicks on it.

Set the ClickEvent

OK, let's make this happen. In the Attributes window for the Button,

find the ClickEvent attribute

click on it to ready it for editing

in the editing field above the list of attributes , type `"next"` (and be sure to include the quotes)

But, Who Is Listening?

Perhaps you are thinking "Great, now this Button can yell 'next'...what does that do for me?"

Absolutely nothing...unless someone is listening. That someone is the Wait object. The Wait object pauses execution of the page in order to listen for incoming event codes (messages). The Wait can then perform an action (such as branching).

Waiting and Branching

Just because the page has a Button does not mean Everest will pause for the user to click on it. You have to tell Everest to wait. That's the purpose of the Wait object: first to pause and then to listen.

Adding a Wait Icon

From the ToolSet

drag a Wait icon (it's the hourglass)

and drop it on the VisualPage editor.

As with the Layout icon, nothing appears in the VisualPage editor. That's intentional. Since there is no visual component for a Wait object, it appears only in the Book Editor.

At run time, the Wait object pauses, waiting for user interaction, until it hears a message worth waking up for. You tell the Wait object what message(s) to listen for.

Editing NextActivator

In the previous topic, you assigned the Button's ClickEvent attribute a value of "next". When the user clicks on the Button, the Button yells "next"; you want the Wait object to listen for this and then do something.

To make this happen, edit the Wait object's NextActivator attribute. In the Attributes window:

click on NextActivator (*not* NextAction)

in the edit field, type `"next"` (including the quotes)

Now the Wait object knows to wake up and do something when it hears someone yelling "next". However it doesn't yet know what to do. This is where the NextAction attribute comes in.

NextAction and BRANCH

Commands, such as BRANCH need not be typed using upper-case letters. However, doing so makes your A-pex3 programs easier to read and share with other authors.

When Wait hears the NextActivator event code (in this case, the word "next") it performs whatever command you type in the NextAction attribute. To branch to the next page, in the Attributes window for the Wait object

click on NextAction

and enter `BRANCH @next`

Be sure to put a blank space between BRANCH and @next.

BRANCH is one of the commands in Everest's A-pex3 programming language...it says "go to" another page in the book. A-pex3 has many more commands, all of which are documented in the technical reference/on-line help.

What's @next?

All special page names in Everest start with the @ symbol. The @start page is the first page in the book. However, unlike @start, @next is *not* an actual, physical page. Instead, @next is a keyword that means "whatever page is next in the book." A BRANCH @next command tells Everest to look at the order of the pages in your book, and branch to the next one, regardless of what that page is actually named.

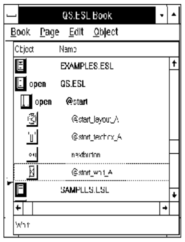
This approach allows you to keep the content of your project separate from its structure, thereby enhancing page reusability.

Via @next (and its cousin @back) you can create a book whose branching is determined simply by the order in which you arrange the pages in the Book Editor. If you later need to insert a new page between two existing pages, it's easy. Simply drag and drop a page icon at the desired location in the Book Editor...there's no need to find and change any branching instructions.

Of course, if you want to branch to a particular page, Everest has the flexibility to handle that as well: you simply specify the name of the desired page in the BRANCH command.

Verify

The arrangement of the icons in the Book Editor should resemble:



Object	Name
	EXAMPLES.ESL
open	QS.ESL
open	@start
	@start_layout_A
	@start_textbx_A
	nextbutton
	@start_wait_A
	SAMPLES.ESL

Wait

If the icons of your page are not in the order shown above (Layout, Textbox, Button, and Wait), correct the order. You can move an icon in the Book Editor by clicking and holding the RIGHT side mouse button on it and dragging it to the desired location.

Save Again

Before continuing, save what you have done so far. Remember how? In the Book Editor

from the Page pull-down menu choose Save.

Taking a Test Drive

OK, your page is now ready to try. In the Author window

from the Run menu choose Preview.

This time the "Preview Finished" message box should not appear immediately. In fact, you may not even notice that the preview has started running.

Click on your page's Next Button

Now you should get the Preview Finished message box...one that also tells you that a branching instruction was encountered. In the Preview Finished message box

click on OK.

Good Work

Great! That's your first page, complete with a tiny bit of user interaction.

The Second Page

Intentionally, the @start page is rather plain. Now, you'll make the window look spiffier with help from the second page.

New Page

To begin creating the second page, from the ToolSet window

drag a Page icon (it's in the upper-left corner of the ToolSet)

and drop it on the VisualPage editor. If your @start page is still open, Everest will close it automatically (but if there are pending changes, it will first ask if you want to save).

When prompted for the name of the new page, enter

```
picture
```

Names can be up to 16 characters in length, but try to stick to eight or fewer characters...shorter page names are easier to deal with if you plan to deliver your project via the Internet.

Add a Textbox

From the ToolSet

drag a Textbox icon

and drop it on the VisualPage editor. To focus on it

click inside the new Textbox, and type

```
SpaceWalk!
```

inside it to act as the "title" for the page.

If you recall, the Textbox on the @start page was not transparent. Let's be different here, and make this one transparent.

Transparency via DrawText

In the Attributes window for the Textbox,

find the Textbox attribute named DrawText

double click on the DrawText attribute

double click on the DrawText attribute again

A combo box is an input field with a drop down list. Click on the arrow at the right of the field to expose the list.

to set it to `-2` Draw. Alternatively, you can click once on DrawText, then choose the -2 Draw option from the Combo box in the Attributes window.

Nothing happens.

That's right, nothing happens because DrawText takes effect only when you run the page. So, run a Preview; in the Author window

from the Run pull-down menu, choose Preview.

Where's my Text?

Is your text invisible now? It may or may not be depending on the default colors in your copy of Windows. If you cannot see the SpaceWalk text, it is because the ForeColor of the Textbox is too similar to the BackColor of the window. Even if you can see the text, you should perform the following steps for the purposes of this Quick Start.

Click OK in the Preview Finished window.

In the Attributes window for the Textbox,

find the ForeColor attribute

double click on ForeColor

to open the Color dialog window.

Click on the bright yellow near the bottom of the Color dialog

click OK.

Drop Shadows

A drop shadow is often used to make text appear more prominent. In the Attributes window for the Textbox

find the DrawShadow attribute

double click on DrawShadow to open the Color dialog

choose the dark yellow/brown color

click OK.

Like DrawText, DrawShadow takes effect only when you run the page. But, don't do so yet. First, make the following changes.

Changing the Font

In the Attributes window

find the FontSize attribute

double click on FontSize to open the Font dialog.

In the Font dialog

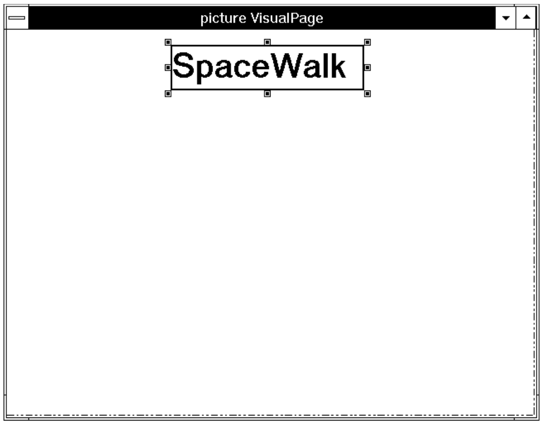
make the font size larger, 24 or so will do

click OK.

In the VisualPage Editor

> resize the Textbox as needed (use the left mouse button to drag the sizing handles)
>
> move the Textbox so it is centered near the top of the VisualPage editor; use the right mouse button (or Shift key + left mouse button) on the sizing handles to drag the entire Textbox to the desired position.

The VisualPage Editor should resemble:



Now, go ahead and run a Preview; in the Author window

> from the Run pull-down menu, choose Preview.

The yellow text should contrast with the background of the window. If it does not, it will at run time, when it will appear on top of the dark, tiled background from the @start page.

> Click OK to close the Preview Finished window.

Adding a Picture Icon

Now, from the ToolSet

drag a Picture (*not* SPicture) icon and drop it on the VisualPage Editor.

Be sure to use the Picture icon, not the SPicture icon.

Loading an Image

At this point, direct your attention to the Attributes window, which should read "picture_picture_A Attributes" at the top. Recall that if the Attributes window is hidden behind others, you can press its hot key, Ctrl+A, to bring it to the top.

In the Attributes window

find the attribute named PictureFile

double click on PictureFile

Everest can load and display many types of graphics files, including: .AVI, .BMP, .FLC, .FLI, .GIF, .JPG, .PCX, .RLE, .TGA, .TIF, and .WMF.

Doing so opens the Load File dialog box. In the list of files, you should see one named spcwalk.gif which was included with Everest as a sample file.

In the Load File window

double click on spcwalk.gif

to tell Everest that's the file you want.

The Load File window will close, and the image of an astronaut should plot in the Picture object. If you cannot see the astronaut, the Picture object may be too small. In the VisualPage Editor,

use your left mouse button to drag the sizing handles of the Picture and make it larger.

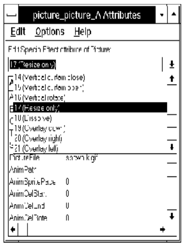
SPCWALK.GIF is a 256-color image. If you are running Windows in a 16-color mode, the colors in the image will be mapped as closely as possible, but may appear blotchy.

Scaling the Image

Unlike the SPicture object, by default the Picture object does *not* scale the image to fit within its box. Fortunately, there's an easy way to do so. In the Attributes window

find the SpecialEffect attribute

use the Combo box drop down list to set SpecialEffect to 17 (Resize only) as shown below



Edit SpecialEffect attribute of Picture:

17 (Resize only)

14 (Vertical curtain close)
15 (Vertical curtain open)
16 (Vertical rotate)
17 (Resize only)
18 (Dissolve)
19 (Overlay down)
20 (Overlay right)
21 (Overlay left)

PictureFile	[illegible]
AnimPath	
AnimSpritePage	0
AnimCelStart	0
AnimCelEnd	0
AnimCelRate	0

The color of the image may degrade momentarily while it is being resized. This is caused by the video display drivers you are using within Windows. Sometimes clicking on the image in the VisualPage editor corrects the color (doing so asks Windows to refresh it). It should be OK when you run the page.

SpecialEffect

Typically, the SpecialEffect attribute is used to display images with a dissolve, wipe, slide or other effect. A side benefit is SpecialEffect resizes the image to fit the Picture object. You probably want to try the effects. In the Attributes window,

> double click on SpecialEffect.

Each time you do so, the SpecialEffect setting advances to the next, and Everest replots the image.

When you are done experimenting,

> set SpecialEffect to 18 (Dissolve)

Picture Transparency

At run time, when this second page appears, we want the image to overlay what is already in the window. That means, part of the image has to be transparent.

TpColor

The TpColor attribute determines which color in the image will be transparent at run time. Wherever you see the TpColor at design time is where the background will show through at run time.

In this case, we want the star field background from the @start page to show through in the area above the astronaut. That area is black. You could manually set TpColor to the color code for black (which is &H0&). But, there's an easier and more reliable way. In the Attributes window

> find the TpColor attribute
>
> double click on it

Doing so opens a message window that tells you to click within the Picture object on the color you want to make transparent. So,

> first click OK to close the message box, then
>
> click inside the Picture on the black area to the right of the astronaut.

Doing so sets TpColor to, you guessed it, &H0&.

But, It's Not Transparent!

As with DrawText for Textboxes, TpColor for Pictures takes effect only when you run the page. Hang on...the page is not yet ready for that.

Help!

If you are beginning to feel a bit overwhelmed by all these attributes, don't worry. Don't try to remember what they all do: Everest has context-sensitive on-line help. Let's give it a try. In the Attributes window

> click once on TpColor, then
>
> from the Help pull-down menu in the Attributes window, choose "Look up TpColor" (alternatively, you can press Shift+F1).

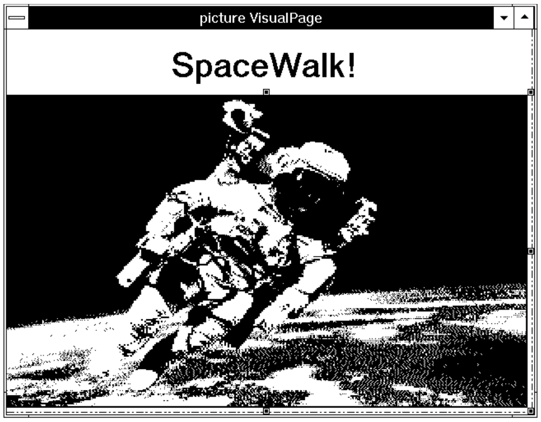
Doing so opens the EVEREST.HLP help file's TpColor entry. Feel free to browse through the hypertext-linked topics. There are over 500 topics and thousands of links.

When done, close the Help:

double click in the Help window's upper-left corner.

Resize the Picture

After closing the help window, you should resize the Picture so it occupies the lower 75% of the window. Leave room for the SpaceWalk! caption at the top. When done, your VisualPage editor should resemble (due to printing limitations, the astronaut image does not reproduce well here)



Object Instancing

One of Everest's most powerful features is object instancing. Object instancing lets you reference an object that was originally created with another page, and incorporate a custom copy of it into the current page.

Saved by the Button

Like the @start page, this page needs a Button object the user can click to proceed. Rather than create the button from scratch, wouldn't it be nice to use the one from the first page? After all, it already has its attributes set perfectly.

Recall that you enabled the SaveAsObject attribute of the Button object on the @start page. This means that Everest saved the Button independently of the @start page, thereby allowing any page to reference it.

Adding a Button

You could add a Button to this page, then "instantiate" the previous one. But, there's a quicker way that saves a step. From the ToolSet window,

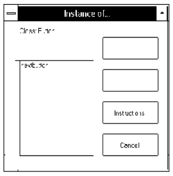
> use the *right* side mouse button to drag a Button icon

and drop it on the VisualPage editor. A menu will pop up.

The right side mouse button tells Everest you want to do something special...that's what triggers the display of the popup menu. From the popup menu

> choose Instantiate/copy.

This opens the "Instance of" window.



Instance of...

Class: Button

nextbutton

The Instance of Window

In the Instance of window, in the list on the left, you should see an item named nextbutton. If the list is empty, it means you failed to enable SaveAsObject for the Button on the @start page. If the list contains something other than nextbutton, it is because you skipped renaming the Button on the @start page. (If you need to go back and correct this, here's how: in the Instance of window, click Cancel. In the Book Editor, double click on the @start page's icon (save changes, if prompted). This opens the @start page. In the Book Editor, click on the Button. To rename, choose Rename from the Edit menu. To enable SaveAsObject, double click on it in the Attributes window. Finally, double click on the picture page's icon in the Book Editor to reload the second page. Save your changes. Drag the Button icon again with the right side mouse button.)

Now, in the Instance of window

double click on nextbutton

to tell Everest you want an instance of that button. Immediately, Everest will create a Button on the page whose attributes are linked to the one from the @start page. To verify this has happened, check the VisualPage Editor, where your Button should now display the same caption (Next) as the one on the @start page.

In fact, Everest updates the Button's location to match that on the first page. This feature is especially handy for navigation buttons like this because you typically want them in the same location on every page.

The Color of the Attributes Window

When working with an object that is shared among pages, Everest alters the background color inside the Attributes window. This is to remind you that changing the attributes here can effect other pages as well. (And, yes, in case you are wondering, Everest has a cross-referencing feature that tells you the names of pages that share an object.)

Instance Attributes

This is going to be the last page in the Quick Start project, so rather than displaying Next on the button, you should display the word End.

Wait! If you change the button's caption here, won't it change on the first page too?

Normally, yes. However, this is a special case. The Caption attribute is what is known as an "instance attribute." Instance attributes are local to the page (i.e. you can safely change them without influencing other pages).

Which are Instance Attributes?

You may have noticed that some attribute names listed in the Attributes window are displayed in bold, and others are not. The attributes in bold are instance attributes.

Since the Button object's Caption attribute is an instance attribute, you can edit it, and you won't change the first page. Go ahead and do so. Change Next to

```
End
```

(by now, you should know the steps for doing so: click on Caption, then type `End`).

There's one more object needed to complete this screen.

More! There's Wait

Finally, drag a Wait icon (the hourglass) from the ToolSet to the VisualPage Editor. Set the NextActivator to

`"next"` (including the quotes)

and the NextAction to

```
BRANCH @end
```

"@end" is a special name that Everest recognizes in a BRANCH command as an instruction to stop running the project.

Save Your Work

Yes, it's time to save the page. Do so (you should know how), then proceed to the next topic that shows how to run both pages.

Running and Editing

In a prior topic, you used the instant preview feature to check the appearance of the

@start page. The preview is an excellent way to view the page you are editing. What about when you want to test more than one page?

Start At...

The Start At... feature on the Author window's Run menu runs a project (one or more pages) as the user will see it. This lets you debug your project, and employ what is perhaps the single most important feature in an authoring system: edit-on-the-fly.

Many authors consider an edit-on-the-fly feature the most important one in an authoring system because it can save so much time.

With an edit-on-the-fly feature, you can pause while test running your project in order to change it (correct a typo, modify a program, change the branching, etc.) and then resume running where you left off. Surprisingly, many authoring systems do not have this feature; in order to make a change, you have to stop running, load the editors, find the problem, fix it, recompile, and start running again from the beginning! That's very time consuming! In fact, Everest's edit-on-the-fly feature works even if your project is stored on an Internet site. In just seconds, you can update your site in real time.

OK, in the Author window

from the Run pull-down menu, choose Start At...

Everest displays a window that shows the location of your project, the book, and the pages. In the Pages column on the right

double click on @start

to begin there.

Running

Your @start page should now be displayed. Before proceeding please note: we've intentionally skipped a step creating the second page. This is so you'll have an opportunity to try Everest's edit-on-the-fly feature. We'll instruct you how. OK, with that warning, to proceed to the second page

click on the Next button.

The Textbox and Picture will be added to the window, but the star field background will not show through around the astronaut.

OK, So Why Didn't it Work?

Via the TpColor attribute, we helped you designate the black area around the astronaut as transparent. Doing so has allowed the background to show through...but, which background is showing?

Here's the problem: the background that is showing through is the background of the Picture object, not the window. Fortunately, this is very easy to correct. You simply tell Everest to copy the background of the window into the background of the Picture object. The following steps tell you how.

Editing on the Fly

The solution is to enable the Picture's CopyBgnd attribute. Let's do that right now, *while still running your project*.

If the main Author window is hidden behind the window in which your project is running, press Ctrl+M to bring it to the top. Then, in the Author window

from the Run menu choose Edit-on-the-Fly

Everest will load the current page (the one named picture) into the editors.

Picture Transparency, Part 2

When enabled (set to Yes) the CopyBgnd? attributes tells Everest to, at run time, first copy whatever is behind the Picture, then load the image on top.

In the Book Editor (press Ctrl+B if it is hidden),

double click on the Picture icon to load in into the Attributes window.

In the Attributes window

find the CopyBgnd? attribute

double click on it to toggle it to `Yes`

Resuming

Now, you can resume running the project where you left off. In the Author window's

from the Run menu, choose Resume and Update

Everest will ask if you want to save changes. You do, so click on

Yes

The Resume feature restores you to the place at which you had invoked the editors. Resume and Update does the same, but also reloads the current page, and runs it again from the start. Use Resume and Update when you make a change to the current page.

Your page should now look correct, with astronaut image overlaid onto the star field background.

The Benefits

How long did it take you to correct the problem with edit on the fly? Perhaps a few seconds? The edit-on-the-fly feature works even if you are test running a project stored on an Internet site. These small changes/corrections are the most common operation when you are testing your project. During the development of an actual project, you'll probably use edit-on-the-fly hundreds of times.

Compare this with what you need to do in other authoring tools. If you were test running your project on the Internet, and found a problem, in other authoring tools you would need to:

> stop the test run
>
> load the authoring program
>
> find the page that contains the problem
>
> make the correction
>
> save the change
>
> recompile, compress, or regenerate your project (perhaps into HTML/Java form)
>
> exit the authoring program
>
> upload the corrected version to the Internet
>
> start the test run again
>
> page ahead to find the place at which you had stopped

That's very time consuming! To review, here's the process in Everest:

> choose Edit-on-the-fly from the pull-down menu
>
> make the correction
>
> choose Resume and Update from the pull-down menu

Fini

When you are done test running, in your project

> click the End Button

to halt it.

Summary

We hope you found this Everest Authoring System introduction helpful. With no programming, and a very small amount of typing, you've created an attractive two-page demo that includes drop shadowed text, special effects, overlaid transparent pictures, buttons and branching.

Yet, we've only scratched the surface of Everest. The system offers many more multimedia features, CBT capabilities, other objects and a sophisticated programming language named A-pex3 (usage is optional).

If you would like to learn more about Everest, consider continuing with the Tutorial. The Tutorial is fairly substantial, and usually requires a few hours to complete all chapters.

We have also appended a Questions and Answers topic here to respond to a few questions many people have upon completing the Quick Start.

Quick Start Questions and Answers

Q. I'd like to run my projects over the Internet. Can that be done with Everest? How fast does it run?

A. Yes. In fact, if you'd like to see the performance for yourself right now, you can do so via Everest's Internet Simulator. You don't even need an Internet connection; Everest will let you choose a simulated modem speed. From the Author window's Run menu, choose Internet Simulator. In the window that appears, choose the book to run (the Quick Start you just created is in the QS.ESL book). Click on the Instructions button there if you need help.

Q. I saw some odd colors in the SPCWALK.GIF picture. Is that normal?

A. Yes. While you are editing, Windows is constantly adjusting its color palette. The odd colors should not appear to the user run time. Also, the SPCWALK.GIF file contains 256 colors. If you are running in a 16-color mode, Windows maps the 256 colors into 16 the best it can.

Q. Since the pages in the Quick Start have similar Wait objects, would it make sense to employ object instancing like we did for the Button object?

A. Yes. Authors frequently use object instancing with Wait objects. Doing so makes it easy to adjust the branching commands for all pages.

Q. Can a group of objects be treated as a unit, and used as a "background"?

A. Yes. That is what the Include object does. You can read more about it in the Design Guide.

Q. What are those dashed lines I see in the VisualPage editor?

A. Those are called Window Size Guides. They help demarcate the size of the run time window. By default, they are set for a 640 x 480 size window, but you can change them as necessary (details are found later in the Tutorial).

Q. Does Everest come with any templates?

A. Yes, a few. The template book is named TEMPLATE.ESL. The best way to employ a template is via the "Load template" feature of the File pull-down menu. After choosing "Load template" navigate your way to the \EVEREST\SAMPLES subdirectory. The list of books should include TEMPLATE.ESL; click on it to open the book. The list of pages should show the names of template pages. Double click on one of the page names.

We intentionally do not package hundreds or thousands of templates with Everest. By adding your own templates, you'll have only those important to you...plus, your project won't end up looking the same as your competition's. If you have several authors working on the same project, consider sharing the template book via a network.

Q. How can I learn more without using the manuals?

A. Almost all the editing windows in the Author program contain an “Instructions” topic (either as a button or pull-down menu item). The on-line help documents every object, attribute and programming command; consider browsing through it for ideas. You might also examine the Everest demo disk. To run it, in the Book Editor, double click on the EVDEMO.ESL book. That opens the book. Next, in the Author window, from the Run menu choose Start at. Then choose the @start page.