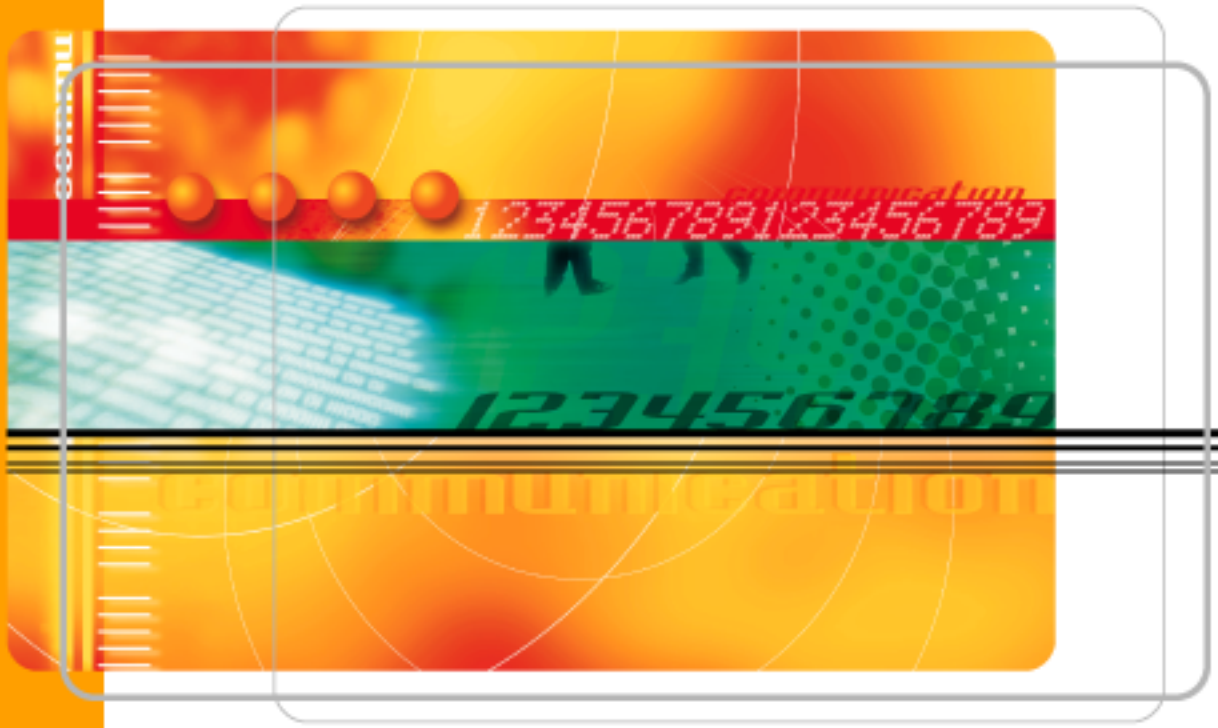


click2learn

User Guide

ToolBook II

INSTRUCTOR™

© 2000 click2learn.com, inc. All Rights Reserved.

The software described in this document is furnished under a Software License Agreement. Please read it thoroughly. The software may be used or copied only in accordance with the terms of this agreement.

Information in this document is subject to change without notice. No part of it may be copied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from click2learn.com, inc.

U.S. GOVERNMENT RESTRICTED RIGHTS

The Software and Documentation are provided with RESTRICTED RIGHTS. If you are an agency of the United States Government, the following clause applies to this license. The Software is commercial computer software developed at private expense. Software provided to the United States Government pursuant to solicitations issued on or after December 1, 1995 is provided with the commercial license rights and restrictions described elsewhere herein. All Software provided to the United States Government pursuant to solicitations issued prior to December 1, 1995 is provided with “Restricted Rights” as provided for in the Commercial Computer Software-Restricted Rights at FAR, 48 CFR 52.227-14 (JUNE 1987) or The Rights in Technical Data and Computer Software clause at DFAR, 48 CFR 252.227-7013 (OCT 1988), as applicable. Use, duplication or disclosure by the United States Government is subject to restrictions as set forth therein. The manufacturer is click2learn.com, inc., 110 -110th Ave. NE, Bellevue, Washington 98004-5840.

ToolBook, OpenScript, and click2learn.com are registered trademarks, and the click2learn.com logo, Ingenium, ToolBook II, ToolBook II Assistant, and ToolBook II Instructor are trademarks of click2learn.com, inc. ToolBook, click2learn.com, and OpenScript are registered in the United States of America and may be registered in certain other countries.

Adobe and Adobe Photoshop are registered trademarks of Adobe Systems, Inc. TrueType is a trademark and QuickTime is a registered trademark of Apple Computer, Inc. InstallShield is a registered trademark and InstallShield Express is a trademark of InstallShield Corporation. Pentium is a registered trademark of Intel Corporation. Macromedia and Flash are registered trademarks and Macromedia FreeHand is a trademark of Macromedia, Inc. Microsoft, Microsoft PowerPoint, Microsoft Visual Basic, Microsoft Windows, Microsoft Windows NT, and ActiveX are registered trademarks, and Microsoft Excel, Microsoft Internet Explorer, Microsoft Paint, and Microsoft Word are trademarks of Microsoft Corporation. Netscape Navigator is a registered trademark of Netscape Communications, Inc. RealMedia is a trademark of RealNetworks, Inc. Java is a trademark or registered trademark of Sun Microsystems, Inc. UNIX is a registered trademark of UNIX Systems Laboratories, Inc. All other product and/or company names are the property of their respective owners.



The ToolBook II Team

Development
Scott Anderson, Tim Barham, Matt Boucher, Arjay Cannata, Lisa Clark, Bryan Comstock, Denny Dedmore, Mark Florence, Michael Hamaoka, Nadim Harfouche, Joel Kittinger, Jeff Krinock, Ian Morrison, Michael Ormes, Claude Ostyn, Matthew Patron, Simon Price, Jan Utterstrom

Quality Assurance
Brian Gordon, Wayne Griffin, Aileen Ibershof, Quan Ton That, Russ Van Rooy, Juan Vera

Publications
Mary Anne Dane, Betty Houser, Julie Kawabata, Jeannie McGivern

Special Thanks
Jason Barbacovi, Brad Crain, Patty Law, Kristin Lucas, Leo Lucas, Steve Martino, Michelle Mazurek, Andrea Mocherman, Raymond Nolan, Jennifer Schlosstein, Paul St. Pierre, Tae Surh, Paolo Tosolini, Paul Wetenhall

Part No. 29901

Publication date: November 2000
Printed and reproduced in the United States of America.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Contents

PART I Welcome to ToolBook II Instructor

Chapter 1
Getting started

Introducing ToolBook II Instructor **20**
Before you install **21**
Installing Instructor **24**
Starting and exiting Instructor **24**
Learning Instructor **26**
About this book **30**

Chapter 2
Basic concepts

About the click2learn.com product family **34**
About ToolBook II Instructor **35**
Planning your project **37**
Building an application **42**
Using the Actions Editor **49**
Using OpenScript **49**
Extending Instructor using Windows technologies **50**
Using resources in your application **53**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 3
Learning the Instructor interface
Introducing the visual interface 56
Working at Author level 58
Working at Reader level 84
Navigating in an application 86

PART II Building your application

Chapter 4
Creating books and pages
About books and pages 92
About templates 94
Creating a new book 98
Creating pages and backgrounds 103
Organizing pages and backgrounds 107
Working with backgrounds 113
Setting the page size 115
Setting properties for books 118
Navigating through a book 119
Enhancing a book's performance 124

Chapter 5

Working with the Catalog

About the Catalog **126**
Authoring with the Catalog **127**
Exploring types of Catalog objects **132**
Customizing the Catalog **137**

Chapter 6

Working with objects

About objects **142**
Adding objects from the Catalog **142**
Creating objects using the tool palette **144**
Selecting objects **147**
Creating hyperlinks **148**
Cutting, copying, pasting, duplicating, and deleting objects **151**
Moving objects **153**
Spreading and centering objects **154**
Flipping and rotating objects **154**
Aligning objects **155**
Changing the size and shape of objects **157**
Modifying the appearance of objects **160**
Moving objects between layers **166**
Grouping objects **172**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 7
Setting object properties

Understanding properties and extended properties **180**
Using the Properties dialog box **181**
Using the right-click menu **185**
Using the Extended Properties dialog box **186**
Using the Property Browser **188**

Chapter 8
Working with text

About fields **192**
Working with fields **194**
Creating fields **195**
Setting field properties **197**
Adding text to fields **201**
Selecting and editing text **205**
Formatting text in fields **209**
Inserting inline graphics **214**
Inserting special characters **216**
Finding and replacing text **218**
Working with record fields **220**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 9
Using graphics

About graphics **228**
Using graphics on the Internet **229**
Adding graphics to your book **229**
Modifying graphics **233**
Creating and applying a color palette **238**

PART III Adding interactive features to your application

Chapter 10
Working with buttons and hotwords

About buttons **246**
Defining how buttons look **246**
Using and modifying buttons **248**
Adding a graphic to a button **258**
About hotwords **263**
Creating and modifying hotwords **264**
Editing hotwords **267**
Removing hotwords **269**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 11
Working with list boxes and combo boxes

About list boxes and combo boxes **272**
Using list boxes or combo boxes on the Internet **274**
Creating list boxes **274**
Setting list box properties **276**
Adding text to a list box **279**
Creating combo boxes **281**
Setting combo box properties **283**
Adding text to combo boxes **284**
Determining user selection **286**

Chapter 12
Creating a quiz using question objects

About question objects **292**
Working with question objects **293**
Question objects in action **297**
Scoring questions **302**
Specifying feedback **304**

Chapter 13
Hiding, showing, and animating objects

Creating visual effects **312**
Hiding and showing objects **312**
Animating Instructor objects **318**
Playing an animation at Reader level **322**
Modifying an animation in the Animation Editor **327**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 14
Using multimedia

About multimedia **332**
Using multimedia on the Internet **332**
Choosing a media player **332**
Using the Universal Media Player **335**
Using other media players **343**
Creating and managing clips **347**

PART IV Adding behavior with action sequences

Chapter 15
Using the Actions Editor

About action sequences **364**
Choosing when to create action sequences **366**
Opening the Actions Editor **368**
Creating action sequences using the Actions Editor **370**
Creating and using shared action sequences **388**
Managing action sequences **396**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 16
Creating action sequences: Practical examples

About the examples **404**
Using the examples **405**
Example 1: Using conditions **407**
Example 2: Using variables **414**
Example 3: Changing object properties **421**
Example 4: Using shared action sequences **426**
Troubleshooting action sequences **441**

PART V Using advanced features

Chapter 17
Using resources

About resources **446**
Creating and modifying resources **449**
Adding resources to a book **452**
Working with resources **454**
Getting information about a resource **461**
Replacing resources **463**
Exporting resources **465**
Removing resources from a book **466**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 18
Importing and exporting data

About importing and exporting **470**
Importing and exporting text **471**
Importing and exporting graphics **484**
Importing books and pages **486**
Customizing the import process **490**

Chapter 19
Creating windows with viewers

About viewers **494**
Creating viewers **503**
Setting viewer properties **508**
Working with viewer properties **510**
Using viewers in an application **520**

Chapter 20
Creating and modifying menu bars

About menu bars **550**
Using the Menu Bar Editor **556**
Working with menu bars **559**
Using OpenScript with menu bars **562**
Handling menu-event messages **576**
Displaying pop-up menus **582**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 21

Using ActiveX, Automation, and other Windows technologies

About ActiveX and Automation **586**
Using ActiveX and Automation in an application **587**
Writing OpenScript for ActiveX controls and Automation objects **591**
Creating action sequences for ActiveX controls **597**
Troubleshooting ActiveX and Automation **600**
About ADO **603**
About OLE **603**
Working with OLE **604**

Chapter 22

Optimizing Instructor applications

Tips for optimizing a book designed for DHTML export **610**
About optimizing OpenScript applications **611**
Optimizing books, pages, and objects **611**
Improving navigation and search speed **615**
Optimizing scripts **618**
Creating bitmap images of pages or backgrounds **625**
Benchmarking performance **630**



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

PART VI Distributing and printing your application

Chapter 23

Preparing applications for release

Organizing and checking your application **636**

Preparing your application based on a distribution method **642**

Using the AutoPackager **647**

Using Instructor on a network **652**

Installing and testing your application **653**

Uninstalling an application **654**

Chapter 24

Distributing applications on the Internet

About Internet distribution options **656**

Choosing an Internet distribution option **657**

Publishing your book to the Web **662**

Using ToolBook II courses with a learning management system **667**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 25
Printing from Instructor

Printing pages **672**
Printing a report **675**
Formatting report text **681**
Previewing pages and reports **683**
Adding and formatting a header or footer **687**
Setting printing conditions **690**
Printing scripts **694**

Glossary 695

Index 733

PART I

Welcome to ToolBook II Instructor

This part of the *ToolBook II Instructor User Guide* describes how to set up and install ToolBook II Instructor,™ introduces basic Instructor concepts, and explores the user interface.

IN THIS PART

Chapter 1 Getting started

Chapter 2 Basic concepts

Chapter 3 Learning the Instructor interface



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 1

Getting started

Welcome to ToolBook II Instructor—the complete courseware authoring system that allows you to create sophisticated online learning applications. In this book, you'll learn about the basic concepts and tools you need to build and manage complete, effective courseware. Before you go on to explore Instructor, read through this chapter. Here you'll learn how to install and start Instructor, find out about the resources available to help you learn Instructor, and preview this book's contents and conventions.

IN THIS CHAPTER

Introducing ToolBook II Instructor **20**

Before you install **21**

Installing Instructor **24**

Starting and exiting Instructor **24**

Learning Instructor **26**

About this book **30**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Introducing ToolBook II Instructor

ToolBook II Instructor is a complete courseware authoring system that allows you to create and distribute online learning applications.

With Instructor, you can focus on what is important: presenting the material to be learned. Instructor helps you create and manage the content, navigation, feedback, and quizzes that are a vital part of your interactive learning applications. You can easily add sound, animation, video, graphics, and other special effects to your applications, making them lively and engaging. Hyperlinks and navigation objects allow your learners to control the direction and pace of their learning experience.

Instructor is designed for use by a broad spectrum of course developers and instructional designers. If you are new to developing online learning applications, you'll find that Instructor's easy-to-use features will get you quickly on your way to producing high-quality interactive courses. More experienced course developers will appreciate the robust content-creation environment of Instructor, the Actions Editor visual programming tool, the powerful OpenScript® programming language, and a variety of other design options.

After you have created a course using Instructor, you can deliver it on demand to any student with access to the Internet, or you can deploy it on your company's intranet. Instructor offers several flexible deployment options. Your students can view your course in a Web browser that displays Internet-based file formats such as HTML (Hypertext Markup Language). Alternatively, students can use the ToolBook II Neuron browser plug-in to view your course as a native ToolBook II book (TBK file) in a Web browser. You can also distribute your course as a run-time (stand-alone) application from a hard disk drive or CD-ROM.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Before you install

Before installing Instructor, take a moment to check your package contents to make sure you received everything.

In addition to this book, the full retail version of Instructor should include:

- a program disc that contains:
  - —the ToolBook II Instructor program.
  - —InstallShield® Express,™ a setup program.
  - —hundreds of media clips, including professional clip art, photos, and video.
  - —a variety of online resources.
- *The click2learn.com Guide to e-Learning*, a resource book about how to plan, create, and deliver interactive learning over the Internet.
- a software license agreement.
- a professional services overview.
- a product registration card.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

System requirements

To run Instructor, your computer configuration must meet the minimum requirements outlined in the table below.

Minimum and recommended system configurations

Component	Minimum	Recommended
Computer	Pentium® 166	Pentium II 233 or higher
RAM (Random access memory)	32 megabytes (MB) for playback; 48 MB for authoring	64 MB or more
CD-ROM drive	Single-speed	Double-speed or faster
Hard disk space	70 (MB)*	Up to 300 MB for all optional components
Operating system	Microsoft® Windows® 95 Service Release 2, Windows 98, Windows NT® 4.0† Windows Millennium Edition, or Windows 2000 Professional†	Same as minimum configuration
Graphics card	Video graphics adapter (VGA) capable of displaying at least 256 (8-bit) colors at 800x600 pixel resolution for playback	VGA capable of displaying 32,768 colors (16-bit) at 800x600 pixel resolution minimum for authoring

* Depending on the options chosen during setup, your installation might require more space.

† You must have an administrator account on Windows NT and Windows 2000 in order to install Instructor. See your system administrator or refer to your Windows NT documentation for details.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Additional multimedia requirements

You can add media to courseware in a variety of ways. Most computers come with the hardware and software components necessary for developing or using multimedia programs. Depending on how you plan to use media in your courseware, you will need some or all of the following items:

- A sound card to play digital audio files
- Speakers or headphones to play audio
- Files required to play digital video, such as video-playback drivers and video codecs (software that compresses and decompresses video files)
- Additional files required to play media or other content on the Internet (for example, browser plug-ins like RealMedia™ and ToolBook II Neuron)



Note You don't need to be a video technician or graphic artist to create compelling courseware for education and training. Many companies now sell images, video clips, and sounds that you can copy into your multimedia application. Usually, the purchase of a company's product that contains media files gives you the right to publish them in your application. Sometimes, however, you must agree to specific licensing requirements. Never assume that you have rights to publish media files; always read the product's documentation. When in doubt, consult the product's manufacturer.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Installing Instructor

To install Instructor and its associated files, use the Setup program, which provides options and instructions for each step. The Setup program decompresses the files and transfers them to your hard disk.

The Setup program is set to run automatically when you insert the Instructor program disc into your CD-ROM drive. You can also manually install Instructor, as described below.

➤ **To install Instructor:**

1. Insert the Instructor program disc into your CD-ROM drive.
2. If the Setup program does not start automatically, from the Start menu, choose Run.
3. In the Run dialog box, type the location from which you are installing Instructor, plus the word *setup*. (For example, type *d:\setup*. If your CD-ROM drive letter is not *d,* substitute the letter of the drive containing your program disc.)
4. Click OK, and then follow the instructions on the screen.
5. After installation is complete, you may be prompted to restart Windows.

Starting and exiting Instructor

You start and exit Instructor as you would any Windows application.

➤ **To start Instructor:**

- From the Start menu, point to Programs, point to ToolBook II, point to Instructor 8, and then choose ToolBook II Instructor.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

When you start Instructor, the ToolBook II Instructor Startup dialog box appears. Use this tabbed dialog box to start a new book or open an existing book.

Tabs organize a variety of choices to help you get started in Instructor.

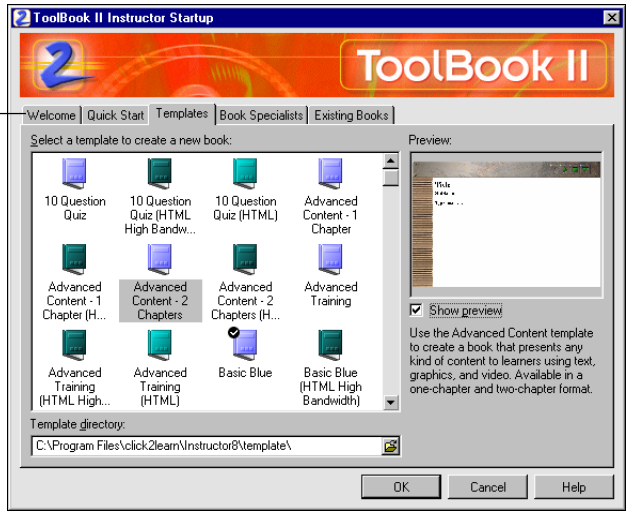


Figure 1
ToolBook II Instructor Startup dialog box

➤ **To exit Instructor:**

◆ Do one of the following:

— If you have not opened a book, in the ToolBook II Instructor Startup dialog box, click Cancel.

— If you have opened a book, from the File menu, choose Exit.



— Click the Close button on the title bar of the ToolBook II Instructor Startup dialog box or the Instructor main window.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Learning Instructor

Instructor includes both online and printed documentation, as well as a number of other resources to help you learn to use Instructor. Further assistance is available from the click2learn.com Web site and from click2learn.com technical support staff.

The Instructor documentation set includes three printed books:

- This book, the *ToolBook II Instructor User Guide*, discusses how to use Instructor's features and provides the context and procedures you'll need to use Instructor effectively.
- *The click2learn.com Guide to e-Learning* provides an overview of the application-development process, including guidance about how to plan the project, work with graphics and multimedia, use the Internet to distribute your courseware, and manage learner interaction.
- *Using the ToolBook Learning Management System* discusses how to work with the learning management feature in Instructor.

In addition, there is an online book available on the program disc:

- *Programming in OpenScript* provides information about how to extend the features of your application using the OpenScript programming language.

➤ **To open an online book:**

- From the Start menu, point to Programs, point to ToolBook II, point to Instructor 8, point to Documentation, and then choose the online book you want.

A variety of other learning resources are discussed in the sections that follow.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the online Help system

The online Help system is a fast, comprehensive information resource about using Instructor. Here, you'll find information on specific dialog boxes, maps of screen elements, keyboard and mouse shortcuts, information about what's new in Instructor, and an OpenScript reference resource. There are several ways to access online Help:

- From the Help menu, choose Help Contents. Then, find the information you need by choosing from the many Help categories and topics.
- Open a context-sensitive Help topic in a dialog box by pressing the F1 key or by clicking the Help button.
- Search on keywords by choosing Search for Help On from the Help menu.
- Display tooltips on the screen by pausing the pointer over a button on a toolbar.

Using the ToolBook II Coach

The ToolBook II Coach is a quick, convenient reference tool that provides context-sensitive information as you're authoring. The Coach displays tips and suggestions about the page you're working on, step-by-step procedures for any object you select, and design guidance. Leave the Coach open as you work and note how the information in the Coach window changes dynamically according to the object and page you're working with.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To display the Coach:**

- From the Help menu, choose Coach.

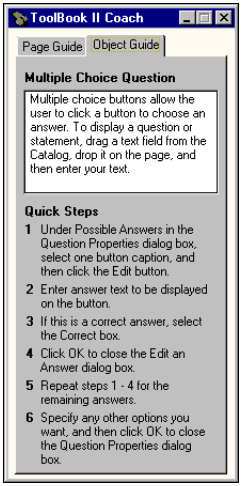


Figure 2
The ToolBook II Coach

Using the resources available from the Developer Corner

The Developer Corner available on the click2learn.com Web site provides one-stop access to a number of valuable tools. You'll find utilities, ToolBook II news, and samples available on the Developer Corner Web page.

➤ **To open the Developer Corner from ToolBook II:**

- Start Instructor if it is not running. From the Help menu, choose Visit the Developer Corner.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Accessing release notes and other files

The release notes (information in a file named RelNotes.wri in the Instructor program directory) are written as the software is completed to ensure that you get the latest possible information about the product. The notes cover exceptions and last-minute information. The notes also provide information on product compatibility, as well as on converting from earlier versions of the product.

Another useful file is the distribution file list (Distrib.wri in the Instructor program directory). This file contains a list of the run-time files required to distribute an application. Both the release notes and the distribution file list are installed along with Instructor and are available on the Instructor program disc.

Visiting the click2learn.com Web site

The click2learn.com Web site offers a variety of information. You'll find technical information and support, see examples of products created with our software, and learn about the many easy solutions click2learn.com provides for creating and managing online learning. Here are some examples from among the many features you'll find at www.click2learn.com:

- Access to the Knowledge Base, a database of product information where you can perform keyword searches for articles
- Access to files that you can download, including sample applications, plug-ins, and updates of current applications, as well as a link to the click2learn.com FTP (File Transfer Protocol) site
- Access to online forums where you can post a question and receive feedback from other ToolBook II authors

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

- Telephone numbers for technical support staff at click2learn.com
- Tips on how best to use click2learn.com products, with creative and diverse examples provided by customers

Contacting click2learn.com for technical support



If you have questions that cannot be answered by the Instructor documentation, online Help, or the online resources available from the click2learn.com Web site, you can contact technical support representatives at click2learn.com. Refer to the topic "Technical Support" in online Help or see the Telephone Support information on the click2learn.com Web site.

About this book

This book has several parts that describe different aspects of building an interactive learning application with Instructor:

Part 1, "Welcome to ToolBook II Instructor," discusses basic Instructor concepts and describes the user interface.

Part 2, "Building your application," includes concepts and techniques to help you construct and customize your online learning courses.

Part 3, "Adding interactive features to your application," describes how to use built-in Instructor features that add interactivity to your application and help you engage your learners.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Part 4, "Adding behavior with action sequences," describes the Actions Editor visual programming tool and provides tutorial-style examples that demonstrate how to use it to add interactive behavior to your courses.

Part 5, "Using advanced features," describes using Instructor's advanced features and integrating other Windows technologies such as ActiveX Data Objects and OLE Automation to customize your courses.

Part 6, "Distributing and printing your application," describes how to prepare and package a course for distribution, how to distribute your course on the Internet, and how to print Instructor pages.

A glossary and an index are also included in this book.

Finding OpenScript instruction

OpenScript is the ToolBook II programming language. Although you can use OpenScript to extend and enhance your courses, you do not need to know OpenScript to use Instructor. Instructor has a sophisticated array of tools, including the Actions Editor visual programming tool, for creating courses without writing scripts. This book provides some OpenScript code examples that you can execute using the script editor or the Command window (click the Script Editor button or the Command Window button on the Instructor toolbar). More information about OpenScript can be found in the online Help system and in *Programming in OpenScript,* an online book available on your Instructor program disc.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Document conventions

This book uses the following visual conventions to help you identify and interpret information.

Document conventions

Example format	Meaning
comboBox	In a paragraph, italic type indicates OpenScript keywords.
Setup	In step-by-step procedures, italic type indicates literal characters to type.
CTRL+ALT+DEL	Keys separated by a plus sign indicate keys to press in combination.
`button id 12`	This font indicates an OpenScript code example.
`\`	A backslash at the end of a line indicates that an OpenScript statement continues on the next line.
<isShift>	In a code example or syntax statement, angle brackets indicate a parameter that represents a literal value or expression.
`--Initialize`	In a code example, two hyphens precede a comment.
`before|after`	In OpenScript syntax, vertical bars separate the options from which you can choose.
`end [step]`	In OpenScript syntax, square brackets indicate optional words and parameters.
...	In OpenScript syntax, an ellipsis indicates that the parameter can be repeated. In code examples, ellipses indicate the presence of any number of statements.
	This icon appears by text that indicates where to look in online Help for more information.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 2

Basic concepts

This chapter provides an overview of ToolBook II Instructor and explains important concepts you should understand in order to use Instructor successfully.

IN THIS CHAPTER

About the click2learn.com product family **34**

About ToolBook II Instructor **35**

Planning your project **37**

Building an application **42**

Using the Actions Editor **49**

Using OpenScript **49**

Extending Instructor using Windows technologies **50**

Using resources in your application **53**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About the click2learn.com product family

ToolBook II Instructor is one of many products offered by click2learn.com, inc. to help you author, manage, distribute, and access interactive online learning applications. Below is a description of some click2learn.com, inc. software tools. For more information about click2learn.com, inc. products and services, visit our Web site at www.click2learn.com.

- **ToolBook II Assistant**™ Designed for ease of use, Assistant includes a rich set of basic draw objects, graphics, and preprogrammed objects with extended capabilities. Its intuitive interface is ideal for those who want to create interactive learning applications quickly—without programming.
- **ToolBook II Instructor** Designed for the flexible creation of content-rich learning applications, Instructor includes the ability to fully customize your application using the Actions Editor visual programming tool and the OpenScript programming language. You can customize objects using these tools and then save them in the Catalog for use in both Instructor and Assistant.
- **Ingenium**™ Designed to advance the management of training, Ingenium offers automated course registration, skills tracking, and facilities management for instructor-led and online courses. This sophisticated training management system can help organizations to streamline administration, share knowledge, and maximize resources.
- **ToolBook II Neuron** Available as a free browser plug-in or ActiveX® control, Neuron allows you to access and run native ToolBook II Instructor and Assistant applications in the Netscape Navigator® and Microsoft Internet Explorer™ browsers. You can optimize the performance of your native ToolBook II applications using Impulse for Neuron, a postprocess utility that segments your book to allow rapid data transfer over the Internet and greatly enhances the speed of delivery.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About ToolBook II Instructor

Instructor is a courseware authoring program that allows course developers and instructional designers to create, customize, and deliver online learning applications. With Instructor, you can create dynamic, interactive courses that engage the learner using sound, animation, video, graphics, and other special effects. Although Instructor provides sophisticated tools for experienced programmers, you don't need to know how to program to use Instructor: You can use convenient prebuilt templates and preprogrammed objects from the Catalog.



Figure 1
The ToolBook II Instructor Catalog

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

With Catalog objects, it's easy to add interactive behavior to your application. You simply drag objects with the behavior you want from the Catalog to a page. The ToolBook II Catalog includes hundreds of objects with preprogrammed capabilities that handle everything from page navigation and quiz scoring to special effects like multimedia feedback and animation. These objects provide prebuilt functionality for the majority of your authoring tasks.

If you are new to developing online learning applications, you'll find that the easy-to-use templates and the Catalog will get you quickly on your way to producing high-quality interactive courses. More experienced course designers will appreciate the robust content-creation environment of Instructor, the Actions Editor visual programming tool, the powerful OpenScript programming language, and a variety of design options.

When you have finished authoring, you can distribute the application you created in several ways: using the Internet or an intranet, a local area network, or a CD-ROM. Instructor prepares your files according to the distribution method you choose.

Exploring the kinds of applications you can create with Instructor

Instructor has specially designed features that allow you to create online courses for training and education environments. You can create a variety of courses with Instructor, including interactive demonstrations, guided tours, and simulations, to name just a few. But you're not limited to courseware development. With Instructor's flexible and versatile authoring environment, you also can create online presentations, kiosk applications, catalogs, and games. The blueprint for your Instructor application resides in your own imagination.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Interactive online learning applications

An interactive online learning application provides computer-based instruction that responds to a student's actions. You can create an Instructor application that not only presents information, but also quizzes students and stores their responses, displays hints, acknowledges correct answers, and tracks progress from lesson to lesson. Details about different types of online learning applications are available in *The click2learn.com Guide to e-Learning,* which is part of your Instructor documentation set.

Planning your project

Good project planning can help you to take advantage of the sophisticated courseware authoring features and powerful distribution capabilities that Instructor offers. A thorough discussion of planning, designing, and evaluating an online learning application is provided in *The click2learn.com Guide to e-Learning*. This section describes specific file organization and distribution issues to consider before you begin using Instructor.

Creating an organized directory structure

Whether you plan to build just one online learning application or several, you'll find it helpful to organize your files. In addition, when you prepare your application for release, you'll want to keep track of any ancillary files that need to be distributed along with your application.

As we'll discuss later in this chapter, applications created in Instructor are composed of one or more files called books. These books should reside in one central book directory, with one or more subdirectories for the media files (sound, graphics, video, and animation) that accompany those books.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Let's say your application has just one book—Introduction to French. We suggest the following directory structure:

```
\French
   IntroFrench.tbk (the ToolBook II book)
   \French media
      EiffelTower.avi (a video file)
```

Now let's say your application is more complex. You have one main book about Romance languages, and several related books about French, Spanish, etc. You could organize the files using several book subdirectories under a central directory, as shown in this example:

```
\Romance
   Romance.tbk
   \Romance and shared media
   \French
      IntroFrench.tbk
      AdvFrench.tbk
      \French media
         EiffelTower.avi
   \Spanish
      IntroSpan.tbk
      AdvSpan.tbk
      \Spanish media
         PradaMuseum.avi
etc.
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Deciding how you will deliver your course

Perhaps the most important decision you'll need to make before beginning an application is how you will deliver your course to your learners: using the Internet or an intranet, a local area network, or a CD-ROM. Because each delivery method may require different preparation, you will want to make this decision early in your project development.

The table on the following page provides a brief overview of application-delivery options and requirements to consider for each method. Chapter 23, "Preparing applications for release," and Chapter 24, "Distributing applications on the Internet," cover these issues in greater detail. You may want to review these chapters before you begin your project so that you are prepared for some of the requirements you'll meet later on.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Prepare your project for release based on a method of distribution

Distribution method	Platform required	Preparation guidelines
Via the Internet or intranet as a series of Web pages that incorporate DHTML	Microsoft Internet Explorer 4.01 SP1 or later; Netscape Navigator 4.06 or later	◆ Do not use the OpenScript programming language in your application; instead, use the Actions Editor visual programming tool. ◆ Ensure that any media or ancillary files are compatible with Internet standards. ◆ Use the ToolBook II Web Specialist to export your application.
Via the Internet or an intranet as a native ToolBook II application displayed in a Web browser using Neuron	Windows 95, Windows 98, or Windows NT 4.0 (or later) operating system; Internet Explorer or Netscape Navigator 3.0 or later	◆ Use all of Instructor's powerful features for creating rich, interactive applications, including the OpenScript programming language and the Actions Editor visual programming tool. ◆ Ensure that users have (or can download) the Neuron browser plug-in. ◆ Use the ToolBook II Web Specialist to prepare your application.

(continued)

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Prepare your project for release based on a method of distribution, *continued*

Distribution method	Platform required	Preparation guidelines
Via a local area network as a native ToolBook II application	Windows 95 SR2, Windows 98, Windows ME, Windows 2000, or Windows NT 4.0 (or later) operating system	◆ Use all of Instructor's powerful features for creating rich, interactive applications, including the OpenScript programming language and the Actions Editor visual programming tool. ◆ If users will install the application on their own computers, use the ToolBook II AutoPackager to create a setup program.
Via CD-ROM or other transferable media	Windows 95 SR2, Windows 98, Windows ME, Windows 2000, or Windows NT 4.0 (or later) operating system	◆ Use all of Instructor's powerful features for creating rich, interactive applications, including the OpenScript programming language and the Actions Editor visual programming tool. ◆ Use the ToolBook II AutoPackager to create a setup program.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Building an application

You build an application in Instructor by starting a book and adding your content: text, interactive objects, graphics, navigation, and so on. You can control the appearance and behavior of objects by setting properties. The Actions Editor visual programming tool can also control the behavior of objects, or you can use the OpenScript programming language in books that will not be exported to DHTML. You create applications at Author level; users interact with them at Reader level.

Author level and Reader level

Instructor has two levels of operation: Author level and Reader level. At Author level, you create and modify applications using authoring and development tools such as templates, the ToolBook II Catalog, dialog boxes, and various palettes. At this level, you can create and modify objects and set properties to define their appearance and behavior.

Reader level is the level at which you test your applications as you develop them, and also the level at which users run your applications. At this level, users can navigate to particular pages, type text into fields, respond to interactive question objects, and trigger events.

For more information about Author level and Reader level, see Chapter 3, “Learning the Instructor interface.”

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

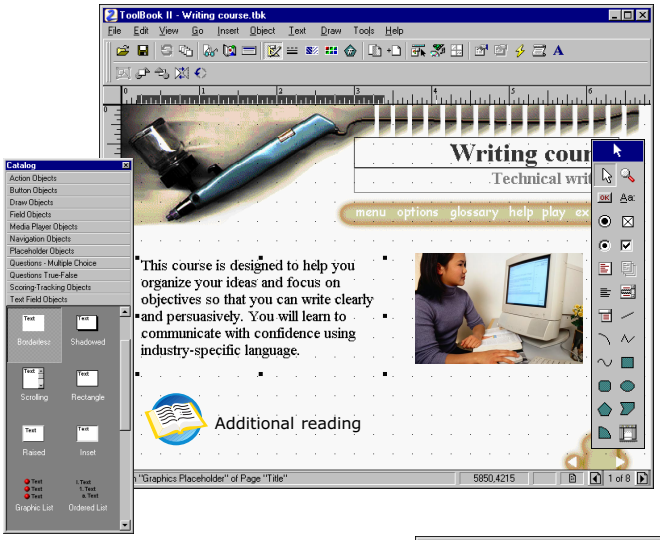
PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



Author level



Reader level

Figure 2
Author and Reader levels in Instructor

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Books and pages

An online learning application created with Instructor consists of one or more files called books. You can order the books to create a sequenced course in which your users access one book after the other. Alternatively, you can join books with hyperlinks so that your users can access them on an as-needed basis; for example, one book may be a glossary for another book in your course.

In your online book, each screen is considered a separate page. Pages contain the elements that determine the look and behavior of your application—buttons, graphics, text, media players, and so on. You arrange the pages in any order that suits your material and then determine a way for your users to navigate through the pages.

A background is the foundation for a page. Several pages can share one background, and a book can contain many backgrounds. Placing objects on the background allows you to maintain a consistent layout throughout your book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Background

Pages that share a background

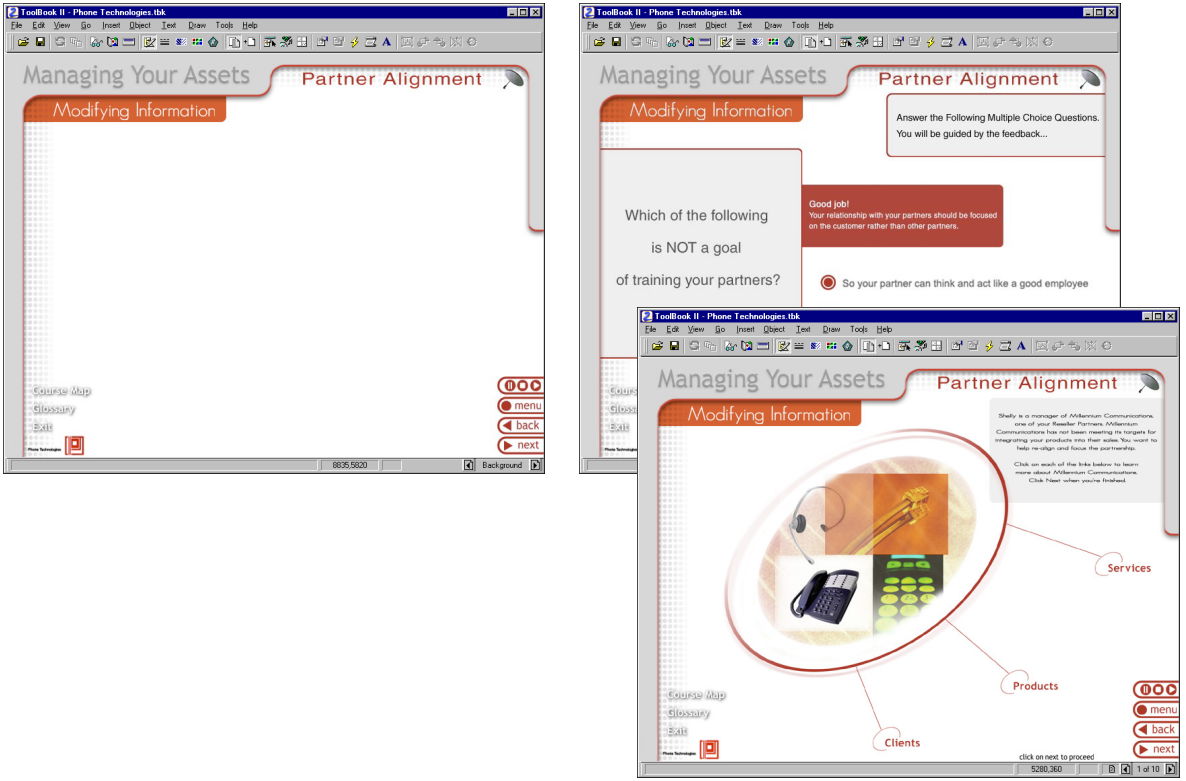


Figure 3
Pages can share a background

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The main Instructor window displays one page. In addition, you can create pages to display in windows that appear along with the main window. These windows are called viewers and can be small, like a pop-up window, or larger, like the main window. For example, you might use the main window to display your application and use a smaller viewer to display glossary definitions.

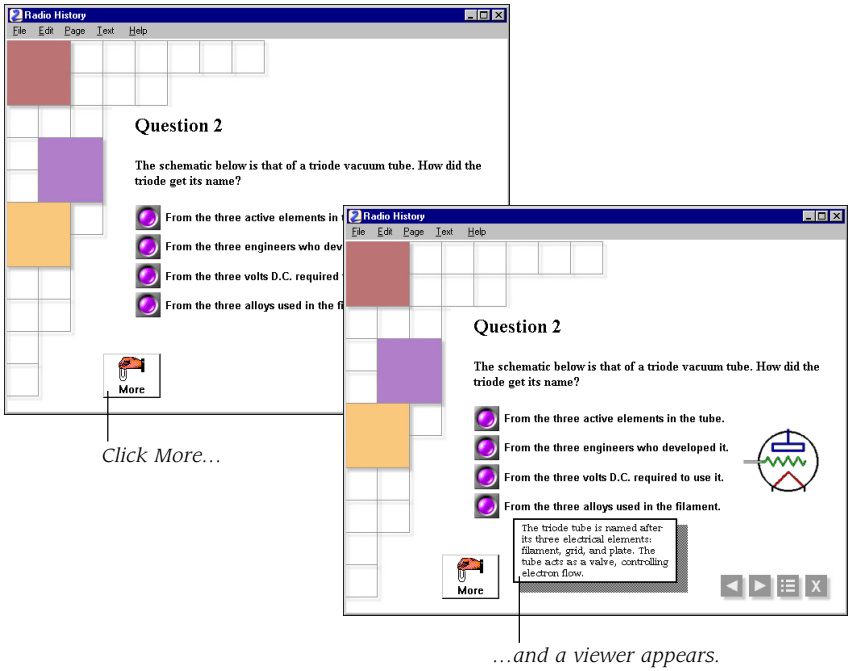


Figure 4
Display multiple pages simultaneously using viewers

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Objects and properties

All of the visual elements of your application—buttons, text, graphics, and even pages and backgrounds—are objects.

You can create objects in Instructor using the tool palette, or you can drag an object from the ToolBook II Catalog and drop it onto your page.

The tool palette contains tools you can use to draw objects on a page. You can draw graphical objects, such as circles and lines, as well as functional objects, such as buttons.

The Catalog contains a rich collection of objects that you use to construct and enhance an interactive application. Many Catalog objects have preprogrammed, built-in behavior.

Each object has a set of properties that define its appearance and behavior. A text field, for example, has properties such as fill color, width, and height that determine its color and size, and other properties such as text alignment and field type that define how it displays text and whether it allows users to enter data.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

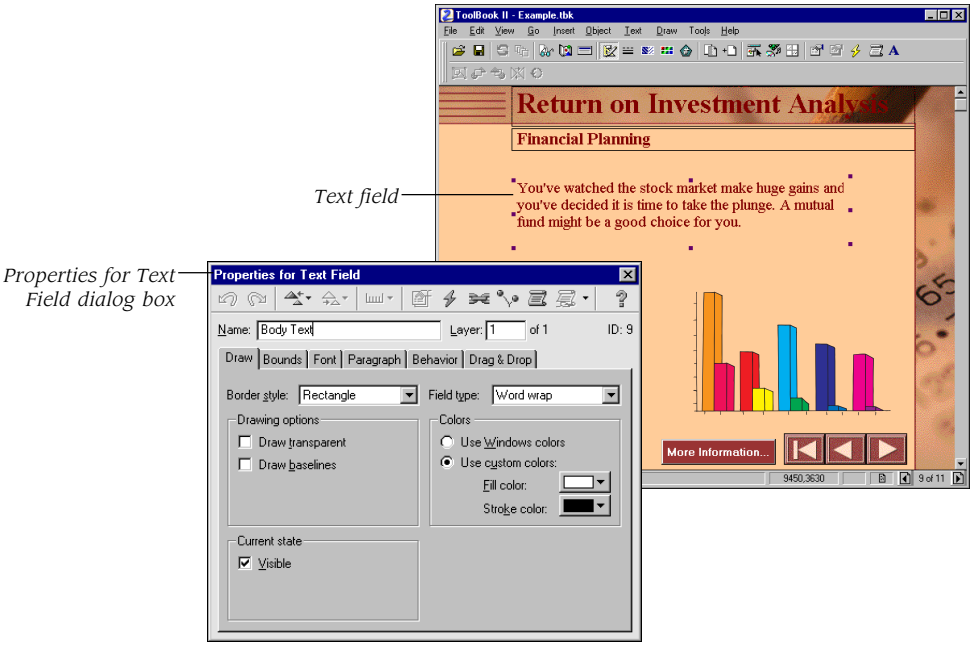


Figure 5
Properties define an object's appearance and behavior

Some objects have extended properties that determine more sophisticated capabilities. A media player, for example, has properties that control the player's interactive functionality, such as which media file will play and how Instructor will play the file at Reader level.

For more information about objects and object properties, see Chapter 6, "Working with objects," and Chapter 7, "Setting object properties."

You can set most properties using dialog boxes available in the Instructor interface. However, some behavior can be set only by using the Actions Editor or OpenScript.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the Actions Editor

The Actions Editor is a visual programming tool that you can use to create sophisticated interactive courses without using OpenScript, the ToolBook II programming language. The Actions Editor is designed to give nonprogrammers a programmer's level of control over object behavior and to provide programmers with a tool they can use to create behavior that will export to the Web.

In the Actions Editor, you use familiar interface elements to create action sequences that play media, display pages or documents, prompt users for information, and much more. In an action sequence, you can specify conditions for behavior or create loops to execute actions repetitively. An action sequence for an object handles a particular event such as a button click, a page loading, or the selection of an item from a list.

For more information about adding behavior to your application using the Actions Editor, see Chapter 15, "Using the Actions Editor," and Chapter 16, "Creating action sequences: Practical examples."

Using OpenScript

OpenScript, the ToolBook II programming language, is easy to use because its syntax is similar to English.

A script you write in OpenScript defines an object's appearance or behavior. For example, a script might control what happens when a learner clicks a button, enters a page, or chooses a selection from a list box. Many of these properties can also be controlled using the Actions Editor or Catalog objects with preprogrammed behaviors; there is often more than one way to accomplish such a task.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Scripts created using the OpenScript programming language do not export to DHTML when you use the ToolBook II Web Specialist. If you've added functionality to your application using OpenScript, you can deliver it over the Internet as a native ToolBook II application using the ToolBook II Neuron browser plug-in, or you can deliver it as a native ToolBook II application over an intranet or on CD-ROM.



For complete information about using OpenScript, refer to the *Programming in OpenScript* online book. You can also consult the online Help system, which contains an OpenScript reference resource.

Extending Instructor using Windows technologies

Instructor provides support for such Windows technologies as ActiveX, OLE, ActiveX Data Objects (ADO), Automation (sometimes referred to as OLE Automation), and Dynamic Data Exchange (DDE). Using these technologies, you can extend Instructor's capabilities and allow Instructor and other Windows programs to share data and interoperate in a networked environment.

ActiveX controls are software components that you can use within an online application to add some type of functionality. These controls allow you to add custom features to a native ToolBook II application or to an application you plan to export to the Web. For more information about using ActiveX in your applications, see Chapter 21, "Using ActiveX, Automation, and other Windows technologies."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

OLE is a technology based on a linking and embedding model that is used to integrate applications. OLE enables you to create an object in one application (the server application) and then incorporate it into another application (the client application). Because OLE objects retain information about where they were created, you can double-click an OLE object and edit it without leaving the client application using a hybrid user interface that includes elements of the server application. For example, you can create a worksheet in Microsoft Excel™ (the server) and bring it into Instructor (the client) using OLE. You can then double-click the OLE object in Instructor and edit the worksheet in place. Using OLE, you can create applications that integrate the capabilities of many different Windows programs. OLE objects are different from ActiveX controls in that an OLE object has a server application that provides its functionality. For more information about using OLE in your applications, see Chapter 21, "Using ActiveX, Automation, and other Windows technologies."

Automation extends Instructor's functionality, enabling you to access and manipulate the objects, properties, and methods of other Windows applications. Automation allows a client application (Instructor) to connect to a server application (for example, Microsoft Word™) by creating an instance of that application as an object. After you have created the application object, you can work with it using OpenScript in the same way you would work with any other object: You can set and retrieve the object's properties and apply its methods. To see an example of how Automation can be used with Instructor, explore the Profiler (Samples\ADO\Profiler.exe on your Instructor program disc). Profiler is a sample application that demonstrates one use of Automation—using Automation to access and manipulate programmable features in Word. For more information about Profiler and Automation, refer to the Profiler online book (Samples\ADO\Profiler.pdf on your Instructor program disc).

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

ADO is a high-level mechanism that allows you to access the data in databases from ToolBook using OpenScript. To see an example of how ADO can be used with Instructor, explore the ToolBook II Instructor Quiz Builder (Samples\ADO\Quizbldr.tbk on your Instructor program disc). The Quiz Builder is a sample application that demonstrates one use of ADO technology—using ADO to send data to and retrieve data from a Microsoft Access database. For more information about the Quiz Builder and ADO, refer to the Quiz Builder online book (Samples\ADO\Quizbldr.pdf on your Instructor program disc).

Dynamic Data Exchange (DDE) is a Windows communication protocol; two programs that support DDE can exchange data and issue commands to each other. For example, Microsoft Excel has a built-in utility for creating charts and graphs, but you can create a more flexible, friendly interface in Instructor. You can maintain data in an Excel spreadsheet and send DDE commands from your Instructor application to produce charts and graphs in Excel. For more information about DDE, refer to Chapter 12, "Using Dynamic Data Exchange" in the *Programming in OpenScript* online book.

Extending OpenScript with dynamic-link libraries

To extend the capabilities of OpenScript, you can call dynamic-link libraries (DLLs), which are libraries of functions available to any Windows program. Windows includes a number of DLLs that you can call to interact directly with Windows; for example, you can display windows, receive Windows messages directly, or determine the current state of the system. In addition, third-party vendors often provide DLLs as a way for you to use their products. For details about how to call the DLLs supplied with another product, refer to the documentation that came with the product.

For more information about DLLs and how to use them to extend the capabilities of OpenScript, see Chapter 13, "Using dynamic-link libraries," in the *Programming in OpenScript* online book.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using resources in your application

You can customize the appearance of your application's interface by importing resources such as cursors, icons, graphics, fonts, menus, and color palettes. For example, you can import:

- a cursor to display under certain circumstances—for instance, when the pointer is paused over an object that cannot be clicked.
- icons or bitmaps to use as graphics on buttons.
- one or more color palettes to change the set of colors Instructor uses on 256-color display devices, creating more realistic bitmaps.

You can create these resources in Instructor or import them from other Windows programs.

For more information about resources and how they can customize your application, see Chapter 17, "Using resources."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



Figure 6
Customize your application's interface with resources

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 3

Learning the Instructor interface

This chapter helps acquaint you with the ToolBook II Instructor visual interface. Here you will learn about each interface component, how to customize the interface, and how to use built-in tools to navigate in Instructor applications.

IN THIS CHAPTER

Introducing the visual interface **56**

Working at Author level **58**

Working at Reader level **84**

Navigating in an application **86**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Introducing the visual interface

Designed for ease of use, the ToolBook II Instructor tools and visual interface let you focus on the important tasks of presenting your content and measuring student results. Instructor has two working levels: Author level and Reader level. At Author level, you build applications using development tools, such as the Catalog, the palettes, and the toolbar. With these tools, you can build books, create and modify objects on pages, and add interactive behavior to your application.

At Reader level, you can test your application to see how it will appear to your users. Users will run your application at Reader level, where they can navigate to pages, respond to question objects, and trigger events. To switch between Author level and Reader level, press F3 or choose Author or Reader from the View menu.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Author level

Reader level

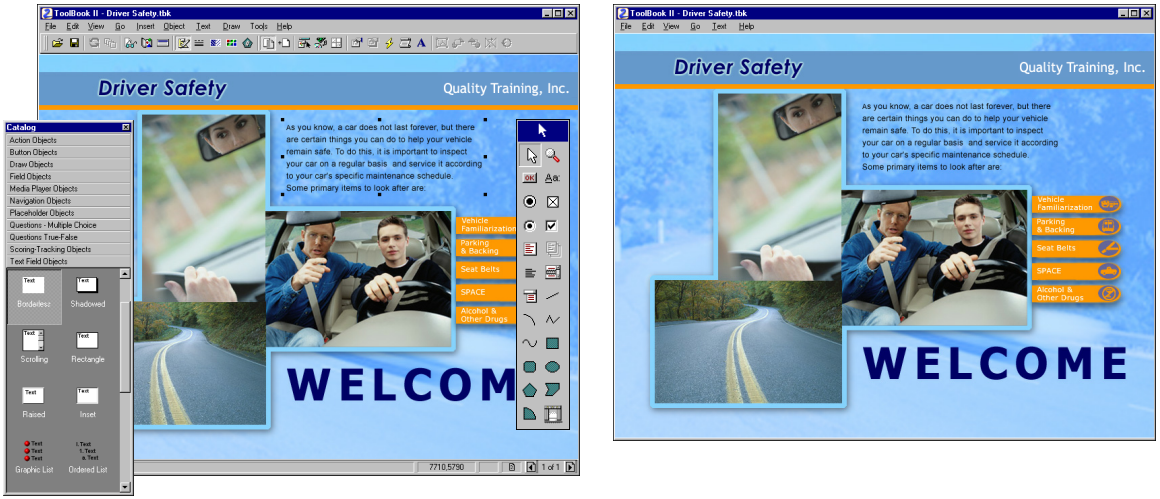


Figure 1
Author and Reader levels in Instructor

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working at Author level

When you first start Instructor, you see the Author-level workspace. At Author level, you use tools such as the menu bar, toolbar, status bar, tool palette, and Catalog to build your application.

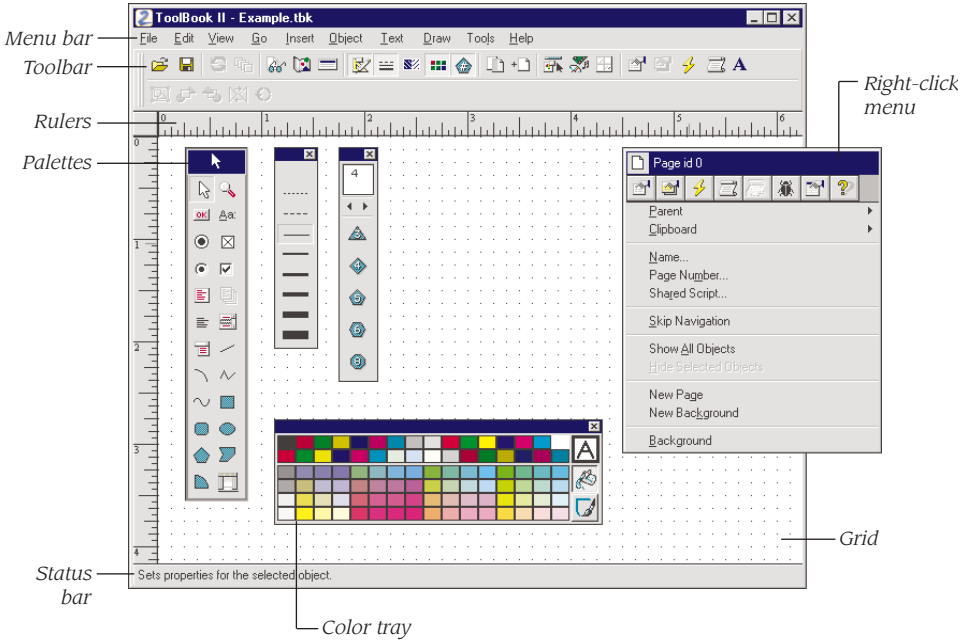


Figure 2
Author-level tools

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The following tools are available at Author level:

- **Menu bar** Allows you to execute commands and access dialog boxes.
- **Toolbar** Provides a collection of buttons that you can use as shortcuts for commonly used menu commands.
- **Status bar** Shows you the name of objects, Help text for menu commands, and the current mouse position. The right side of the bar provides a quick way to select the current page and contains tools that you can use to navigate between pages.
- **Palettes** Contains collections of tools or buttons that you use to draw new objects or change an object's shape, color, or other properties.
- **ToolBook II Catalog** Contains graphics, navigation panels, question objects, media players, and other preprogrammed objects that you can use to design applications.
- **ToolBook II Coach** Provides authoring guidance and step-by-step instructions as you work.
- **The Properties dialog box and Extended Properties dialog box** Allow you to specify properties that determine an object's appearance and behavior.
- **The Actions Editor** Allows you to add interactive behavior to your Instructor applications that will export to DHTML.
- **Browsers** Allow you to view and work with your book in a nonlinear way by organizing your book by properties, handlers, objects, or pages.
- **Rulers** Show you the exact size and position of objects.
- **Grid** Displays a matrix of dots that helps you size and position objects precisely.
- **Right-click menus** Appear when you right-click an object, providing quick access to common settings and dialog boxes.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can hide and show each of these tools and customize their appearance. The following sections describe in detail each interface element's function and capabilities.

Working with the menu bar

You work with the Instructor menu bar as you would work with any Windows menu bar. Simply click the menu name and choose a command from the menu. You can modify how the menu bar appears at Reader level; for more information, see Chapter 20, "Creating and modifying menu bars."

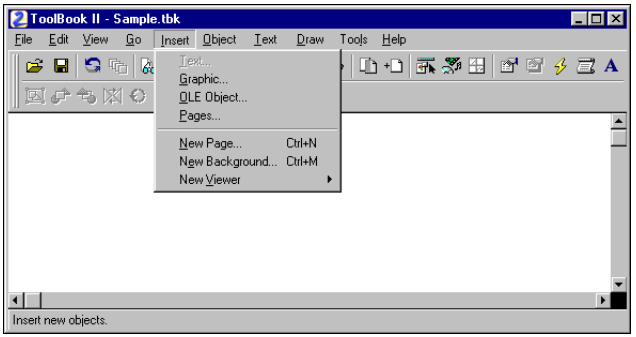


Figure 3
The Author-level menu bar allows access to commands and dialog boxes

Working with the toolbar

The toolbar provides quick access to common menu commands and editing tools. Some of the positions on the toolbar contain two buttons: a default button and a hidden button. To display hidden buttons, press CTRL.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Toolbar command buttons and their function

Default toolbar button	Toolbar button when CTRL is pressed
 Open another ToolBook II application	 Run another application or instance of Instructor
 Save your book	 Save your book under a different name
 Undo last action	
 Duplicate selected object	
 Show or hide the Object Browser	 Show or hide the Page Browser
 Show or hide Catalog	
 Show or hide Command window	
 Show or hide tool palette	
 Show or hide line palette	 Show or hide line ends palette
 Show or hide pattern palette	
 Show or hide color tray	
 Show or hide polygon palette	
 View current background	
 Add new page to your book	

(continued)

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Toolbar command buttons and their function, *continued*

Default toolbar button	Toolbar button when CTRL is pressed
 Open Resource Manager dialog box	
 Open Clip Manager dialog box	
 Open Viewers dialog box	
 Open Properties dialog box	
 Open Extended Properties dialog box	
 Open Actions Editor	
 Open script editor	 Open script editor for shared scripts
 Open Font tab of Properties dialog box	 Open Paragraph tab of Properties dialog box
 Group or ungroup objects	
 Bring to front	 Bring object closer
 Send to back	 Move object farther
 Flip horizontally	 Flip vertically
 Rotate left	 Rotate right

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To hide and show the toolbar:**

- From the View menu, choose Toolbar.

Working with the status bar

The status bar, which displays information about Instructor and the current page, has several parts:

- **Caption area** Displays Help text for menu commands, the Help text for toolbar or tool palette buttons, or the name of an object, depending on the position of the pointer. Instructor also displays progress messages in the caption area when a process is under way.
- **Mouse position indicator** Displays the coordinates of the pointer in page units.
- **Page selection indicator** Shows that you are working on the foreground. You can click here to select the entire page.
- **Navigation control buttons** Move you to the next or previous page when clicked.
- **Status box** Displays the current page number and total number of pages, or the word "Background" if you are working on the background.



Tip By default, Instructor displays the status bar only at Author level. However, you can display the status bar at Reader level if you want to use it to display information. At Reader level, the status bar does not include the page selection indicator. To display the status bar at Reader level, press F12.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

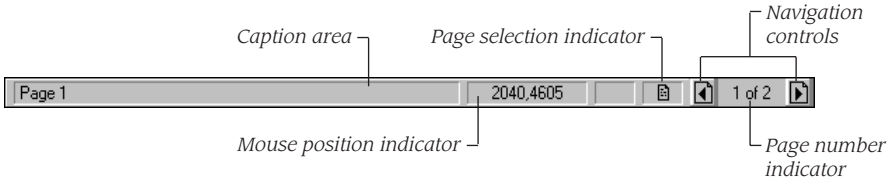


Figure 4
The status bar at Author level

If you create viewers, which are additional windows, each viewer can have its own status bar. You can hide and show each status bar individually, including the one for the main window.

➤ **To hide or show the status bar:**

Using the interface

- From the View menu, choose Status Bar. Alternatively, press F12.

Using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

- To show or hide the status bar for the current window, use the *show* or *hide* command. To specify that a window should show or hide the status bar by default, set its *authorStatusBar* or *readerStatusBar* property:

```
--Displays status bar in current window
show statusBar

--Removes status bar from current window
hide statusBar

--Shows status bar at Reader level
readerStatusBar of viewer ID 0 = true
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with palettes

A palette is a collection of tools or buttons that you use to draw new objects or to set values that modify existing objects. For example, you choose a tool on the tool palette to draw a new object, and you choose a line width on the line palette to determine how thick a line should be. Instructor includes six palettes, described in the following table.

Instructor palettes

Use this palette...	If you want to...
 Tool	Create new objects on a page.
 Line	Set line width or border around an object.
 Line ends	Add arrow heads or tails to a line.
 Pattern	Set a pattern or design to fill an object.
 Color	Set the color of an object or of text.
 Polygon	Set the number of sides, from 3 to 99, of the next polygon to be drawn.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

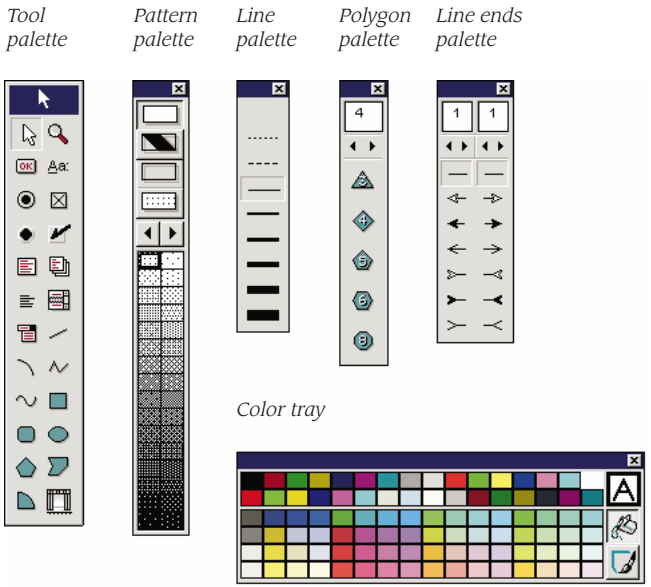


Figure 5
Instructor palettes

By default, Instructor displays only the tool palette. However, you can show as many palettes as you like, hide them to make room onscreen, or move them around as needed.

➤ **To hide or show a palette:**

Using the interface

- From the View menu, point to Palettes, and then choose a palette. Alternatively, click the palette's button on the Instructor toolbar.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- Use one of these statements to show or hide a palette:

```
show linePalette
hide polygonPalette
visible of toolPalette = true
send line
```

➤ **To move a palette:**

- Position the pointer at the top of the palette, and then drag the palette to its new position.

Working with the Catalog

The Catalog contains the authoring objects you use to build your interactive online learning application. There are hundreds of objects in the Catalog, including question sets, media players, navigation panels, and other interactive objects, many with preprogrammed behavior that you can customize. To include an object in your application, you simply drag it from the Catalog to your page or background. Then, set its properties or extended properties to further define its appearance and behavior. In addition, you can link an action or sequence of actions to any Catalog object using the Actions Editor or write a script for an object using OpenScript.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



Figure 6
The ToolBook II Instructor Catalog

➤ **To display the Catalog:**

◆ Do one of the following:

— From the View menu, choose Catalog.



— Click the Catalog button on the Instructor toolbar.

— Press F2.

For more information about using Catalog objects to build an application, see Chapter 5, “Working with the Catalog.”

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with the ToolBook II Coach

The ToolBook II Coach is a convenient reference tool that provides context-sensitive information as you're authoring. The Coach displays tips and suggestions about the page you're working on, step-by-step procedures for any object you select, and design guidance. Leave the Coach open as you work and note how the information in the Coach window changes dynamically according to the object and page you're working with.

➤ **To display the Coach:**

- From the Help menu, choose Coach.

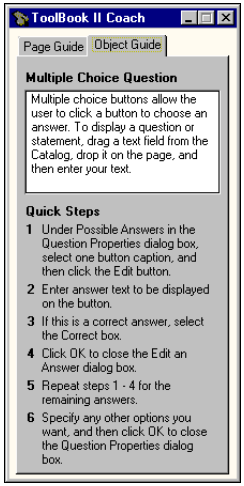


Figure 7
The ToolBook II Coach

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with the Properties and Extended Properties dialog boxes

After you add an object to the page, you set its properties to customize its appearance and behavior in your application. You set an object's properties by first selecting the object and then opening the object's Properties or Extended Properties dialog box. These dialog boxes allow you to set an object's color, specify a graphic displayed on a button, identify the correct answers in a question object, control when a video plays, and much more.

Working with the Properties dialog box

Using the Properties dialog box, you simply enter or select the values you want to set. As a convenience, you can leave the Properties dialog box open as you work—the Properties dialog box shows you the properties of the currently selected object. This feature allows you to quickly set the properties of different objects on your page without having to close and reopen the dialog box several times.

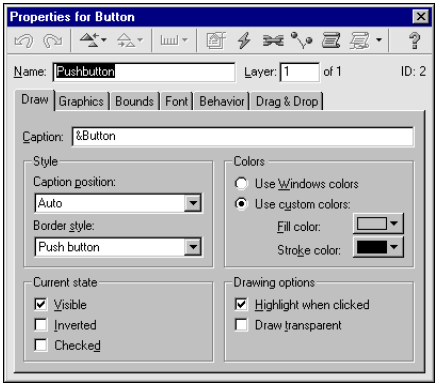


Figure 8
The Properties dialog box

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The toolbar in the Properties dialog box includes buttons to give you quick access to commands, other object properties, and online Help. For example, if you make a mistake, you can click the Undo button to reverse the changes you made.

You can also use the Properties dialog box to specify behavior that extends beyond individual objects. For example, you can set properties for a page, a background, and a book. Instructor also provides a Properties for Lesson dialog box, where you can set scoring and logging options.

➤ **To display the Properties dialog box:**

◆ Do one of the following:

— From the Object menu, choose one of the properties options.



— Click the Properties button on the Instructor toolbar.



— Right-click an object, and then click the Properties button on the right-click menu toolbar.

— Press SHIFT+F6.

Working with the Extended Properties dialog box

Many objects from the Catalog have extended properties that define their behavior. In addition to the Properties dialog box, these objects have an Extended Properties dialog box, where you can specify and refine these extended capabilities.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To display an object's Extended Properties dialog box:**

◆ Do one of the following:

— With the object selected, from the Object menu, choose the extended properties command. (For example, if a true/false question is selected, choose Properties for True/False Question.)



— With the object selected, click the Extended Properties button on the Instructor toolbar.



— In the object's Properties dialog box, click the Extended Properties button on the dialog box toolbar.

— With the object selected, press CTRL+F6.

For more information about setting object properties, see Chapter 7, "Setting object properties."

Working with the Actions Editor

The Actions Editor is a visual programming tool that you can use to build or edit a series of behaviors called an action sequence. In the Actions Editor, you use familiar interface elements, including menus and a toolbar, to construct sophisticated behaviors based on standard programming protocols. Using the sequences you create in the Actions Editor, you can respond to an event (a button click, for example) by playing media, prompting users for input, changing the appearance and behavior of objects, controlling user navigation, and much more. You can also specify conditions for behavior, insert loops to execute actions repetitively, and set and get variables—all without writing a script.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Action sequences are a good way to deliver custom interactive behavior on the Web. By using the ToolBook II Web Specialist, action sequences can be exported automatically to DHTML.

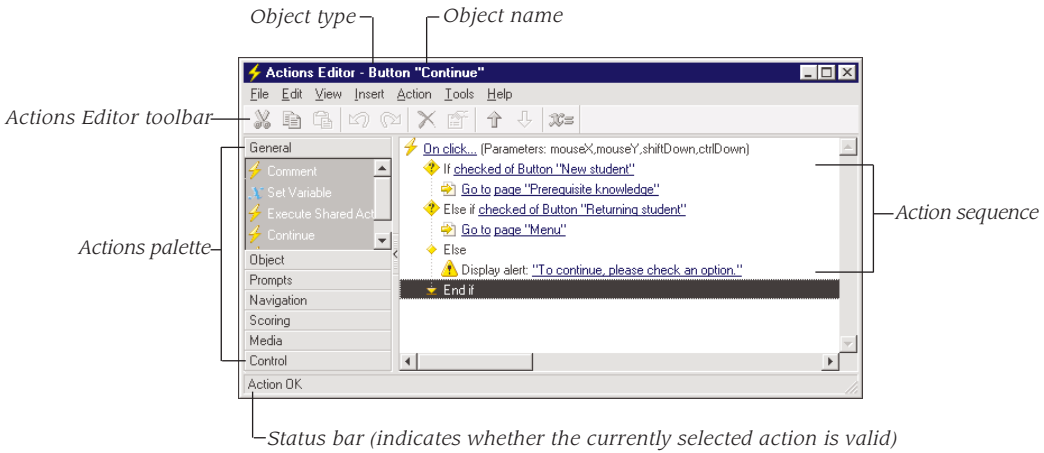


Figure 9
The Actions Editor window

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To open the Actions Editor for an object:**

◆ Do one of the following:

— With the object selected, from the Object menu, choose Actions.



— With the object selected, click the Actions button on the Instructor toolbar.



— Right-click the object, and then click the Actions button on the toolbar of the right-click menu.

— With the object selected, press F5.



— With the object selected, display the Properties dialog box, and then click the Actions button on the toolbar of the Properties dialog box.

For more information about the Actions Editor, see Chapter 15, "Using the Actions Editor," and Chapter 16, "Creating action sequences: Practical examples."

Working with the ToolBook II browsers

Instructor includes four browsers that display information and allow you to set or edit values. They are the Property Browser, the Object Browser, the Page Browser, and the Handler Browser. Each of these browsers is useful for a different set of tasks.

➤ **To open one of the browsers:**

◆ From the View menu, point to Browsers, and then choose one of the browsers.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The Property Browser

The Property Browser displays and allows editing of all available properties for an object. Similar to the Properties dialog box and the Extended Properties dialog box, the Property Browser is a place to view and edit an object's properties. However, the Property Browser displays an object's properties using the OpenScript property name. You should be familiar with OpenScript before using the Property Browser to edit an object's properties.

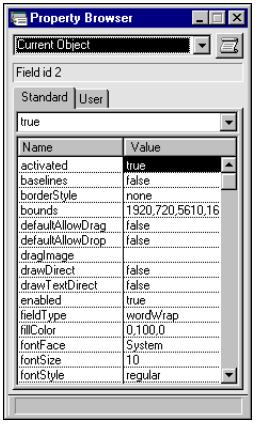


Figure 10
The Property Browser

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The Object Browser

The Object Browser displays and allows editing of all objects on the current page and background. You can edit the properties of an object displayed in the Object Browser by selecting the object and then clicking the Properties or Extended Properties button on the Object Browser toolbar. The Object Browser is useful for editing objects on complex pages, where it may be difficult to select an individual object.

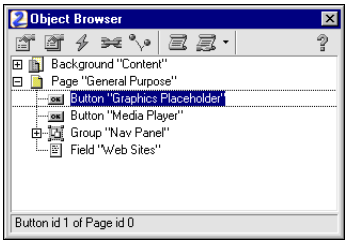


Figure 11
The Object Browser

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The Page Browser

The Page Browser allows you to view and organize the pages in your book. You can view and edit information about an individual page, such as the page's name, number, and hyperlinks. The Page Browser is also useful for reordering pages in your book—simply drag and drop the page icons into a new order.

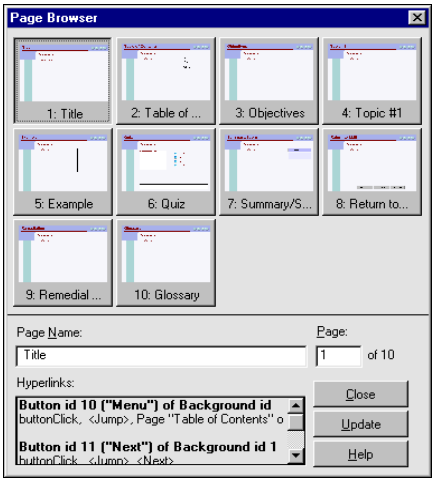


Figure 12
The Page Browser

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The Handler Browser

The Handler Browser displays and allows editing of handlers in the message-passing hierarchy for the current object. You can use the Handler Browser to edit the OpenScript for an object by double-clicking a handler. The script containing that handler will open with the selected handler highlighted. You should be familiar with OpenScript before using the Handler Browser.



Figure 13
The Handler Browser

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with rulers and the grid

You can use the rulers and the grid to help you precisely size and align objects. The rulers appear along the top and left edges of the main window unless you hide them to make more room in the window. As you move the pointer inside the window, a line appears in each ruler indicating the position of the pointer. If you select an object, shadows in the rulers indicate the object's dimensions.

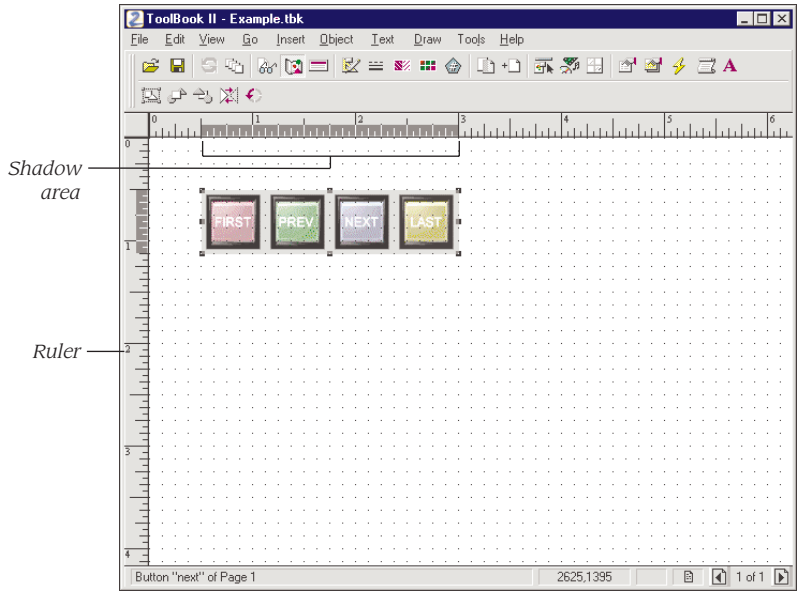


Figure 14
The rulers in the main window

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To hide or show the rulers:

◆ From the View menu, choose Rulers. Alternatively, press CTRL+R.

By default, Instructor displays the rulers using the units of measurement specified in Windows. You can change the units of measurement within Instructor in the Grid dialog box.

The grid is a matrix of dots spaced an equal distance apart in the main window. You can specify that Instructor align objects with the grid automatically when you move or resize them. The grid does not have to be visible for you to "snap" objects to it.

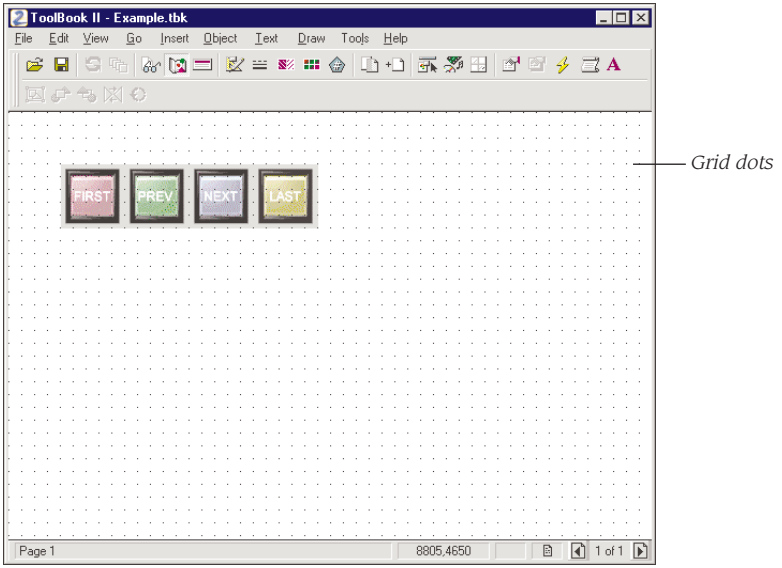


Figure 15
The grid in the main window

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can show and hide the grid as needed at Author level; however, even if the grid is visible at Author level, it never appears at Reader level. You can set the grid spacing to any width you want.

➤ **To change grid settings:**

1 From the View menu, choose Grid.

2 In the Grid dialog box, do one or more of the following:

- To show or hide the grid, select or clear the Show grid check box.
- To change the default space between grid dots, enter an increment in the Page units, Pixels, or Inches (Centimeters) box.
- To specify whether objects align with or move independently of the grid, select or clear the Snap to grid check box.
- To specify the units of measure, select Metric or English.

3 Click OK.

Working with right–click menus

If you pause the pointer over an object on the page and click the right mouse button (right-click), Instructor displays the object's right-click menu. The right-click menu provides an easy way to set the object's properties and provides information about the object's name and related objects. Each object's right-click menu is slightly different, but all display the type of information illustrated in Figure 16.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

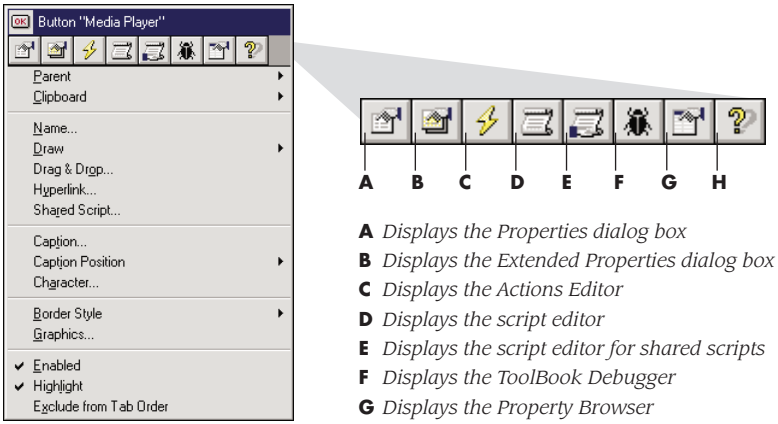
PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



A *Displays the Properties dialog box*
B *Displays the Extended Properties dialog box*
C *Displays the Actions Editor*
D *Displays the script editor*
E *Displays the script editor for shared scripts*
F *Displays the ToolBook Debugger*
G *Displays the Property Browser*
H *Displays context-sensitive online Help*

Figure 16
Sample right-click menu and the right-click menu toolbar

➤ **To display an object's right-click menu:**

◆ Click the right mouse button according to the table on the following page.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Right-click menu access

Object type	Where to click
Page	Right-click an empty spot on the page, or CTRL+right-click anywhere on the background.
Background	CTRL+right-click on the page, or right-click an empty spot on the background.
Viewer	CTRL+SHIFT+right-click anywhere in the viewer.
Book	SHIFT+right-click anywhere on the page.
Other objects	Right-click the object.

By default, you can display right-click menus only at Author level. However, you can change the Instructor default to display right-click menus at Reader level as well. Using right-click menus at Reader level makes it easy for you to change an object's properties while you are running or testing an application.

➤ **To enable right-click menus at Reader level:**

Using the interface

1 From the View menu, choose Options.

2 On the Interface tab, select the Show right-click menus at Reader level check box.

To disable right-click menus at Reader level, click to clear this check box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- Use the following statements to enable right-click menus at Reader level:

```
--At Reader level
sysReaderRightClick = true
send readerRightClick              --If enabled, will disable
```

Working at Reader level

You switch to Reader level from Author level to run or test an application. When you do, Instructor removes the tools you use for developing an application.



Tip The quickest way to switch between Author level and Reader level is to press F3.

Specifically, Instructor makes these changes:

- The Author-level menu bar changes to a Reader-level menu bar that does not contain development-related commands.
- The toolbar, Catalog, Coach, palettes, rulers, dialog boxes, and grid disappear.
- The status bar and right-click menus disappear, though you can choose to make them available.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To switch between Author level and Reader level:**

◆ Choose Author or Reader from the View menu, or press F3.

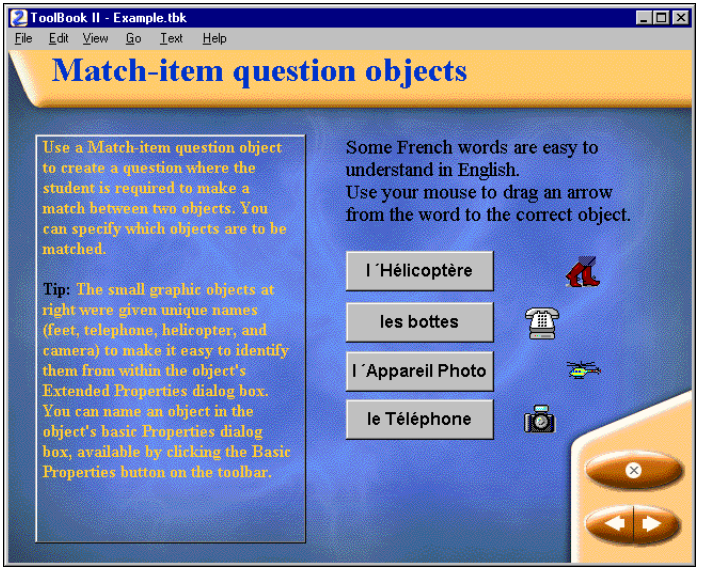


Figure 17
Reader level

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Navigating in an application

You can use a variety of methods to move from one page to another in a book, including:

- page navigation features on the status bar.
- commands on the Go menu.
- arrow keys.

Using the status bar page navigation features

You can use the page navigation features to navigate at Author level.

➤ **To navigate using the status bar:**

- Do one or a combination of the following:
  - — Click the right arrow or the left arrow to go to the next or previous page.
  - — Click the numbers. Instructor will prompt you to enter the number of a page. Click the check mark or press ENTER to go to that page. Click the X or press ESC to cancel.

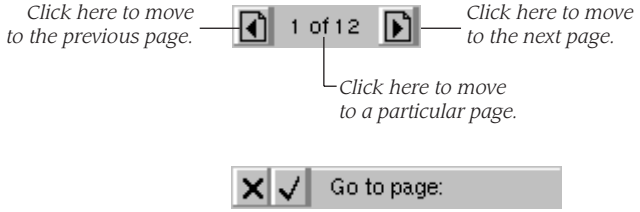


Figure 18
Using the status bar to navigate

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the Go menu and arrow keys

The following table outlines how to use commands on the Go menu and arrow keys to move between pages in a book.

Moving from page to page

To move to this page...	From the Go menu, choose...	Or press these keys...
Next	Next Page	ALT+RIGHT ARROW
Previous	Previous Page	ALT+LEFT ARROW
First page in book	First Page	ALT+UP ARROW
Last page in book	Last Page	ALT+DOWN ARROW
Last page visited	Back	SHIFT+F2

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

1 Getting started

2 Basic concepts

3 Learning the Instructor interface

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

PART II

Building your application

This part of the *ToolBook II Instructor User Guide* presents concepts and techniques to help you construct and customize your online learning application. In these chapters, you will learn about starting a new book, using the Catalog, and adding elements such as text and graphics.

IN THIS PART

Chapter 4 Creating books and pages

Chapter 5 Working with the Catalog

Chapter 6 Working with objects

Chapter 7 Setting object properties

Chapter 8 Working with text

Chapter 9 Using graphics



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 4

Creating books and pages

In this chapter, you will learn about creating books and pages, setting book properties to specify how your book will run, setting up navigation between pages, and controlling how objects display on a page. In addition, this chapter provides some tips about enhancing the performance of your book.

IN THIS CHAPTER

About books and pages **92**

About templates **94**

Creating a new book **98**

Creating pages and backgrounds **103**

Organizing pages and backgrounds **107**

Working with backgrounds **113**

Setting the page size **115**

Setting properties for books **118**

Navigating through a book **119**

Enhancing a book's performance **124**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About books and pages

The ToolBook II Instructor applications that you create and run are called books. A book is composed of pages that contain objects. Objects that you place on a page are displayed on top of a background. A background is the foundation for a page that is shared by any number of pages in a book. One book can contain several backgrounds, each designed for a specific purpose.

Objects that you place on a page's background appear in the same position, style, and size on every page that shares that background. For example, you might put a panel of navigation buttons on the background so that they appear in the same place on every page. Making changes to several pages this way is easy: If you change the background of any page, you see the changes on all of the pages that share the background. You will also save disk space by placing objects on the background, because Instructor stores only one copy of each background and the objects on that background.

When you want an object to appear on only a single page, put it on the page itself (the foreground).

➤ **To switch between the page (foreground) and background:**

- Choose Foreground or Background from the View menu, or press F4 to toggle from one to the other.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

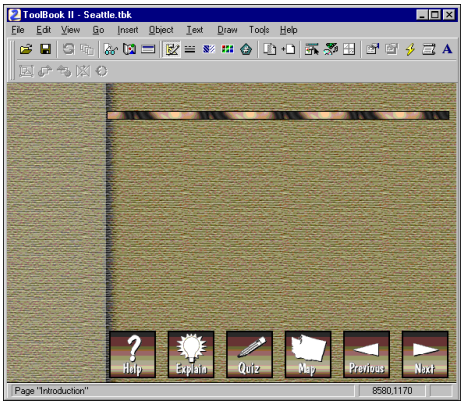
PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Background



Pages that share a background

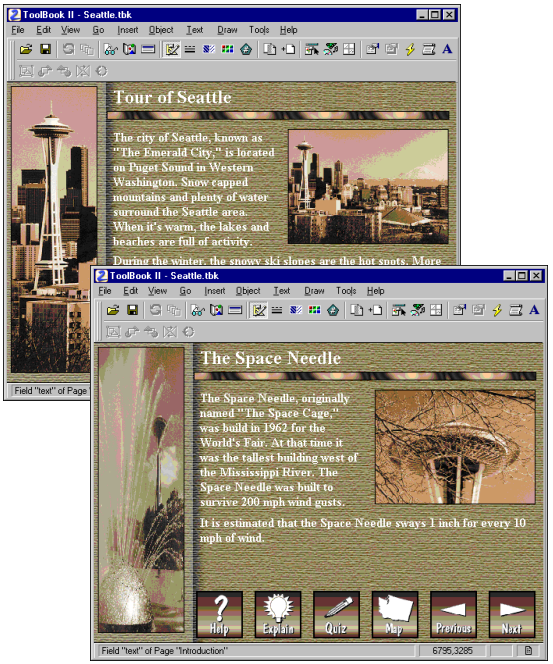


Figure 1
Pages that share a background

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can display pages in a book's main window and also in special viewers—like a pop-up window. For details about creating and using viewers, see Chapter 19, "Creating windows with viewers."

About templates

A template is a prebuilt book that provides a basic structure for your application, including page organization, graphic theme, interactive objects, and navigation controls. To this structure, you add your own custom content. You can fully customize templates by adding or deleting pages, changing page order, or replacing graphics. Using templates can help you save time and provide a consistent look and feature set for each new application.

You can choose a template from the New Book dialog box (choose New from the File menu) or from the ToolBook II Instructor Startup dialog box that appears when you open Instructor. Instructor provides a variety of special-purpose and general-use templates that you can select in these dialog boxes, and you can create your own custom templates that can appear here as well. When you select a template and click OK, Instructor opens a copy of the template that you name and save.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

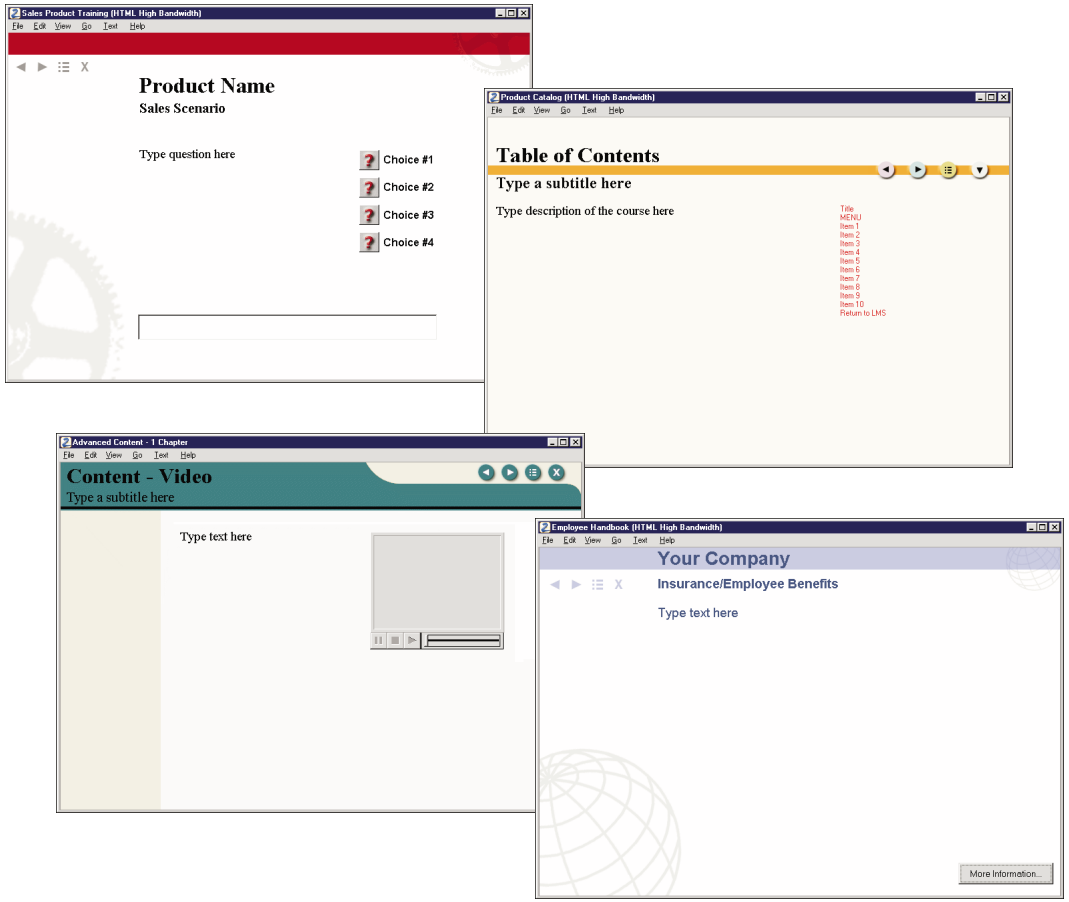


Figure 2
Instructor templates

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

When selecting a template, consider your plan for distributing the application. Some templates are available in different versions, each identified by a distinguishing icon color on the Templates tab of the ToolBook II Instructor Startup dialog box and the New Book dialog box. If you plan to distribute your application on the Internet after exporting to DHTML, select the HTML version of the template optimized for use on the Internet. If your users have a high-bandwidth Internet connection (for example, on a corporate intranet), select the version of the template optimized for high-bandwidth Internet connections.

Designating a default template

Using a default template, you can create a consistent look and feel across multiple online learning applications. A default template can serve as the enterprise-wide standard, giving all of your courseware a unified appearance and organization. For example, if you want to use the same background graphics, number of pages, and general layout for all online learning applications developed in your organization, you can distribute the same default template to all of your courseware authors.

You can use the default template included with Instructor, or you can designate any template to serve as the default template. The default template appears on the Quick Start tab of the ToolBook II Instructor Startup dialog box and the New Book dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To designate the default template:**

1 From the View menu, choose Options.

2 In the Instructor Options dialog box, click the New tab.

3 Under the option labeled Use template for new books, enter or navigate to the book you want to use as the default template.

A shortcut for the template appears on the Quick Start tab. A small check mark on the icon indicates that the template is the default template.



Tip You can create a shortcut on the Quick Start tab for any template on the Templates tab. Simply right-click a template icon and then choose Show Shortcut on Quick Start Tab.

Creating and saving templates

In addition to using the variety of templates that Instructor provides, you can create your own custom templates by saving any book as a template. Books that you create from scratch, books that you create using one of the Instructor templates, and books that you create using one of the ToolBook II Book Specialists can all be saved as templates.

When you save a book as a template, you'll want to save it in the Template directory of the Instructor program directory. That way, when you start Instructor or choose New from the File menu, the custom template you've created appears on the Templates tab of the ToolBook II Instructor Startup dialog box and the New Book dialog box. The Template directory is the default location for storing templates. If you choose another directory in which to store templates, you can navigate to that directory using the Templates directory list on the Templates tab.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To save a book as a template:**

1 From the File menu, choose Save as Template.

2 In the Save as Template dialog box, enter the template title, description, file name, and directory, and then click OK.

After developing a template, you may want to provide guidance about the template to other ToolBook authors who will use the template to create applications. To describe the purpose of a page in a template or guidelines for its use, you can write descriptive text that will appear in the ToolBook II Coach when the page is selected. For more information about assigning your own descriptive text, refer to "Adding Coach text to a custom Catalog object," in the *Customization Guide* online book available from the Developer Corner on the click2learn.com Web site.

Creating a new book

You can start a new book in Instructor by using a template, creating a blank book from scratch, or using one of the ToolBook II Book Specialists. When you start a new book you'll need to choose one of two possible ways to deliver the book: as a native ToolBook TBK file or as Web pages that automatically incorporate DHTML features. Based on your choice, certain menu commands and options may be unavailable if unsuited to the selected deployment method. For more information about your delivery options, see Chapter 23, "Preparing applications for release," and Chapter 24, "Distributing applications on the Internet."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Starting a new book using a template

You can save time and provide a consistent look and feature set for each new application by using a template. As described in "About templates," earlier in this chapter, a template serves as the architecture for your content. When you use a template to start a new book, Instructor opens a copy of the template, which you name and save.

➤ **To start a new book using a template:**

1 Do one of the following:

- If Instructor is not running, start Instructor.

  The ToolBook II Instructor Startup dialog box appears.

- If Instructor is already running, from the File menu, choose New.

  The New Book dialog box appears.

2 Click the Templates tab, select a template icon, and then click OK.

  The Confirm Deployment Method dialog box appears; click Yes to continue.

3 In the Select a Name and Location for Your New Book dialog box, name the file, accept the default directory or choose another file location, and then click OK.

  Instructor opens the book.



Tip You can bypass the Instructor Startup and New Book dialog boxes and automatically open the default template each time you open a new book. In the Instructor Options dialog box (available from the View menu), clear one or both of the following check boxes: Show startup dialog and Show New Book dialog when starting new book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Instructor provides a default template to help you get started quickly. The default template appears on the Quick Start tab of the ToolBook II Instructor Startup dialog box and the New Book dialog box.

➤ **To start a new book using the default template:**

1 Do one of the following:

- If Instructor is not running, start Instructor.

  The ToolBook II Instructor Startup dialog box appears.

- If Instructor is already running, from the File menu, choose New.

  The New Book dialog box appears.

2 Click the Quick Start tab, select the Default Template icon, and then click OK.

  The Confirm Deployment Method dialog box appears; click Yes to continue.

3 In the Select a Name and Location for Your New Book dialog box, name the file, accept the default directory or choose another file location, and then click OK.

  Instructor opens the book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Starting a new, blank book

If you choose not to use a template, you can start with a blank book with no preprogrammed objects or graphic theme.

➤ **To start a new, blank book:**

1 Do one of the following:

- If Instructor is not running, start Instructor.

  The ToolBook II Instructor Startup dialog box appears.

- If Instructor is already running, from the File menu, choose New.

  The New Book dialog box appears.

2 Click the Quick Start tab, select either the Blank DHTML Book icon or the Blank Native ToolBook Book icon, and then click OK.

  The Confirm Deployment Method dialog box appears; click Yes to continue.

  Instructor opens the book.

Starting a new book using the Book Specialists

The ToolBook II Book Specialists automate the task of creating the layout for an Instructor application. The Book Specialists are wizards that gather input from you about how you want the book to appear and behave. Instructor uses this input to build a book based on your specifications.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Instructor offers three Book Specialists to help you build a new book:

- **Full Specialist** Builds a book that reflects your input about distribution method, page styles, and outline for page organization. You can design a sophisticated outline of pages that can serve as the basis for a complete electronic textbook.
- **Lesson Design Specialist** Builds a book that reflects your input about instructional goals, page styles, and learning topics. You can include prepared text and graphics files as well as preprogrammed question objects.
- **Quick Specialist** Builds a book that reflects your input about a few select settings. The Quick Specialist is a streamlined version of the Full Specialist.

➤ To start a new book using one of the Book Specialists:

1. Do one of the following:
   - If Instructor is not running, start Instructor.

     The ToolBook II Instructor Startup dialog box appears.
   - If Instructor is already running, from the File menu, choose New.

     The New Book dialog box appears.
2. On the Book Specialists tab, select the Full Specialist icon, the Quick Specialist icon, or the Lesson Design Specialist icon, and then click OK.

   The Book Specialist opens.
3. Follow the instructions in the Book Specialist.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating pages and backgrounds

Pages and backgrounds are Instructor objects that you can create and modify. In an Instructor book, you can:

- add and remove new pages and backgrounds.
- import pages and backgrounds from a different book.

Adding pages and backgrounds to a book

You can add a new page to your book by choosing the New Page or New Background command from the Insert menu. While both commands allow you to add a page to your book, they each give you different options about the type of page or background you add.

Choosing New Page from the Insert menu allows you to add an empty page using the current background, duplicate a page from your book, or add a page from a template. After choosing one of these options, you can specify that the page introduce a new background, so that changes made to this new background will not affect the background you duplicated.

Choosing New Background from the Insert menu allows you to add a page that uses a new background. You can add a new page with an empty background, duplicate a background from your book, or add a background from a template. Choosing New Background always creates a background with a unique identification (ID) number.



Note You do not need to switch to the background of a page to add a new background. However, you do need to switch to the background to place objects on the background. To switch to the background, press F4.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To add a new page:

Using the interface

1 Go to the page after which you want to add the new page.

2 Do one of the following:

- To add a new page that uses an existing background, choose New Page from the Insert menu, or press CTRL+N.

  The New Page dialog box appears. Enter a name for the page, and then select an empty page, a page from the current book, or a page from a template.

- To add a new page that uses a new background, choose New Background from the Insert menu.

  The New Background dialog box appears. Enter a name for the background and a name for the page. Select an empty background, a background from the current book, or a background from a template.

3 Click OK.

Instructor inserts the new page after the current page (the page displayed in the Instructor main window) and gives it the next sequential page number.



Tip You can modify your settings so that Instructor adds new pages without opening the New Page dialog box. From the View menu, choose Options; on the New tab, clear the Show New Page dialog on New Page command check box. Now Instructor will automatically apply the settings from the page you are on to the new page it creates. You can disable the New Background and New Book dialog boxes in the same way, and Instructor will automatically create a new background or book based on your default selections.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

- To add a new page that uses an existing background, display a page that uses the background you want, and then send the *newPage* message from the page:

```
send newPage to system
```

- To add a new page that uses a new background, display the page after which you want to add the new page, and then send the *newBackground* message:

```
send newBackground
```

Instructor inserts the new page after the current page (the page displayed in the Instructor main window) and gives it the next sequential page number.

Importing pages from another book

You can import pages from another book, called the source book, into the current book, called the target book. Importing pages into a target book does not affect the source book. You can import all of the pages from a source book or only specific pages. Instructor attempts to merge text information into the correct location and adds the resources on the imported pages to the resources of the target book. More information about resources is available in Chapter 17, "Using resources."



Tip Importing pages is especially useful when you want to include a range of pages from another book. If you want to add a single page from another book, you can use a simple copy-and-paste operation.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

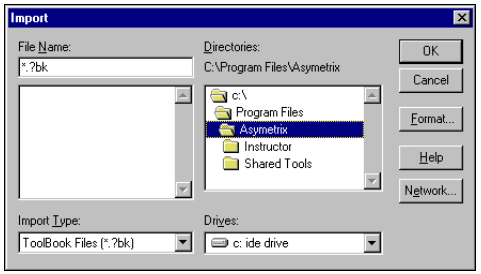
➤ To import pages:

Using the interface

1 Open the target book, and then display the page after which you want to import the source book's pages.

2 From the Insert menu, choose Pages.

Instructor displays the Import dialog box.



3 In the Import type list, select ToolBook II files (*.?bk).

4 In the File name box, select the source book or type the source book's name, including its path if the book is in another directory.

5 To specify a range of pages, or to specify whether the background is imported, click Format. Enter a range of pages, select or clear the Include background check box, and then click OK.

By default, Instructor imports all pages with their backgrounds from the source book unless you specify otherwise.

6 Click OK.

Instructor imports the specified pages from the source book into the target book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- To import the entire source book into the target book, display the page after which you want to import the book, and then use the *import book* command:

```
import book "c:\toolbook\add.tbk"
```

- To import one or more pages, use the *import pages* command. To import pages without the source book's background, add the *without background* parameter. For example:

```
import pages 5 to 5 of book "c:\toolbook\add.tbk"
import pages 12 to 24 of book "c:\toolbook\new.tbk"
   without background
```

If the source book's page size is larger than that of the target book, you may need to increase the target book's page size to see all of the objects. For details, see "Setting the page size," later in this chapter.

Organizing pages and backgrounds

You can reorder your book's pages in a variety of ways. You can use the Page Browser to drag and drop your pages into a new order, or you can renumber, delete, or cut and paste your pages into order.

Using the Page Browser

You can use the Page Browser to examine and modify the overall organization of the pages of your book. The Page Browser displays a thumbnail (miniature image) of each page, as well as information about the current (selected) page, including its name, page number, and hyperlinks.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the Page Browser, you can:

- view and modify the name and page number of a page.
- view a list of hyperlinks for a page.
- reorganize the order of the pages in a book by editing the page number or by using a drag-and-drop operation.
- navigate to a specific page by double-clicking the thumbnail.

➤ To open the Page Browser:

- From the View menu, point to Browsers, and then choose Page.

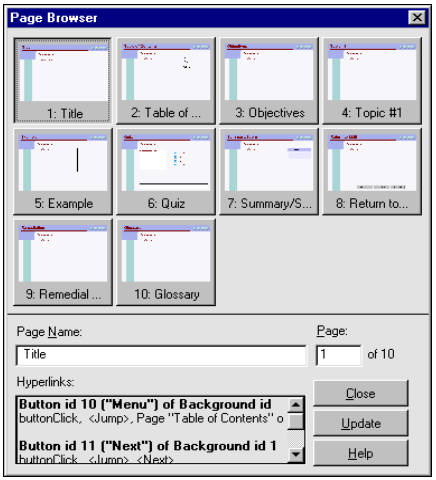


Figure 3
The Page Browser

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Viewing and modifying page information

The bottom portion of the Page Browser shows you information about the page that corresponds to the thumbnail you have selected. For example, after you select the thumbnail of a page, you can view hyperlink information and modify the name and number of the page. You can also use a drag-and-drop operation to change the order of the pages in your book.

➤ **To modify a page's name or number:**

1 Open the Page Browser, and then select the thumbnail of the page you want to modify.

2 Do one or more of the following:

- Enter a new name for the page in the Page name box.
- Renumber a page in one of the following ways:

  —Enter a new page number in the Page box.

  —Drag the thumbnail to a new location.

3 Click Update to apply the changes to your book.

4 *Optional*. Close the Page Browser.



Tip You can also modify the name or page number of a page by right-clicking the page and then choosing the Name command or the Page Number command from the right-click menu.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Renumbering pages in a book

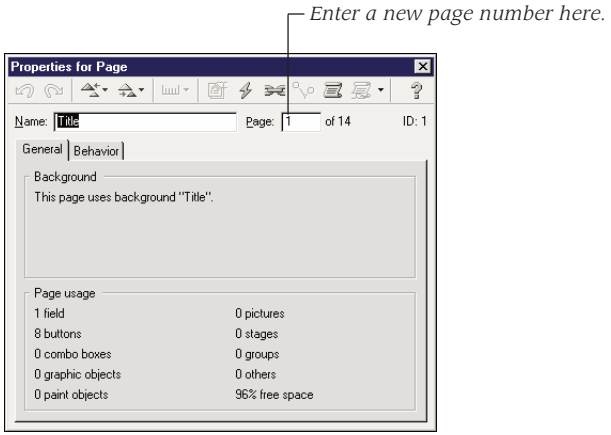
The preceding section describes how you can use the Page Browser to reorder pages. You can also use the Properties for Page dialog box, OpenScript, or a cut-and-paste operation to renumber pages in your application. Instructor removes the page from its old position in the page order and inserts it at the new position. The other pages are renumbered to reflect the change. The page keeps its original background wherever you move it in the book.

➤ **To change the order of a page in a book:**

Using the interface

1 Go to the page you want to renumber.

2 From the Object menu, choose Properties for Page to open the Properties for Page dialog box.



3 In the Page box, enter a new page number.

4 *Optional.* Close the Properties for Page dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- Set the *pageNumber* property to renumber a page and change its order in a book, where <*value*> is the new page number:

```
pageNumber of this page = <value>
```

You can also move a page by cutting and then pasting it. This method is especially useful if you want to move a page to a different book.

➤ To cut and paste a page:

1. Go to the page you want to cut.
2. Choose Select Page from the Edit menu, or click the page selection indicator on the status bar.
3. From the Edit menu, choose Cut.

   Instructor removes the page.
4. In the current book or in another book, go to the page after which you want to paste the page you cut.
5. From the Edit menu, choose Paste.

   Instructor inserts the page after the current page and renumbers the pages in the book.



Note When you reorder the pages in your book using a cut-and-paste operation, Instructor gives the page you are moving a new ID number. When you reorder the pages in your book using either the Page Browser or the Properties for Page dialog box, the ID numbers do not change.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Removing pages and backgrounds

You can remove a page and its background from a book. However, a background that is shared by more than one page in the book remains in the book as long as one page that uses that background remains in the book. To entirely remove a background from a book, you must remove every page that shares that background. When you remove the last page that uses a background, Instructor removes the background as well.

➤ **To remove a page and background from a book:**

Using the interface

1. Go to the page you want to remove.
2. Choose Select Page from the Edit menu, or click the page selection indicator on the status bar.
3. Do one of the following:
   - From the Edit menu, choose Cut.
   - From the Edit menu, choose Delete Page.
   - Press DELETE.

   The Cut command removes the page from the book and puts it on the Clipboard; the Delete Page command and the DELETE key remove the page entirely.

Using OpenScript

- Use the *select* command and the *cut* or *clear* message to remove a page from a book:

```
select this page
send clear
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with backgrounds

Using backgrounds to group like pages together can save you time and reduce the size of your file. You can specify the color, pattern, or backdrop of a background by setting the background's properties. If you decide to make a change, simply modify the background and the change appears on all the pages that share the background.

Applying color and a pattern to a background

The background provides a common design for all pages that share that background. You can create a consistent look and interesting visual effects by adding color and a pattern to your background.

➤ **To apply a color and pattern to a background:**

Using the interface

1. From the Object menu, choose Properties for Background, and then click the Draw tab.
2. To apply a color, under Colors, choose Use custom colors. Click the Fill color button to choose a fill color from the pop-up palette or click Other to open the Color dialog box, where you can specify a custom color.
3. To apply a pattern, under Pattern, choose a pattern style from the pop-up palette or click Other to open the Choose Pattern dialog box. Choose a stroke color to use in the pattern.
4. *Optional.* Close the Properties for Background dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- You can set properties for a background by specifying the background's name or ID number, or you can set properties for the current background by using *this background*. To set colors, set the *fillColor* and *strokeColor* or the *rgbFill* and *rgbStroke* properties. To set the pattern, set the *pattern* property:

```
fillColor of this background = green
pattern of background "Intro" = 25
```

Adding and removing backdrops

You can use a backdrop to make your backgrounds more interesting. A backdrop is a bitmap that appears on a background. You can specify how the bitmap will appear by choosing a style—such as tiled, stretched, or centered—for the backdrop.

Bitmaps used as backdrops are automatically added to your book as resources. If you delete a backdrop from a page, the resource used by the backdrop might remain in the book. See Chapter 17, "Using resources," for more information about resources.

To add or change a backdrop:

- **Using the Catalog** You can add or change the backdrop of a background by dragging a backdrop or bitmap from the Catalog to a page. When you drag a backdrop or bitmap to a page, Instructor displays the Graphic Options dialog box, where you can specify how to apply the graphic and what sizing behavior it should use.
- **Using the Properties for Background dialog box** You can also add and remove backdrops on the Draw tab of the Properties for Background dialog box. To add a backdrop, click Choose Backdrop, select or import the bitmap you want to use, and specify the style. To remove a backdrop, click Clear Backdrop. You can create transparent areas in a bitmap by selecting the Use chromakey option and specifying a color value.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



Note If you are setting a property for a bitmap resource, you need to set it only once for the resource. Instructor automatically applies the property everywhere the resource is used.

Setting the page size

You can specify the size of both the book and the background using either the Properties dialog box or OpenScript. A background's page size takes precedence over the book's page size. The Page tab in the Properties for Book dialog box displays the default page size setting; when you change the book size, the change is reflected in any background's page size that hasn't been explicitly set and in all new backgrounds that you create.



Note When you export your book to DHTML, the ToolBook II Web Specialist will size the browser window according to the size of the largest background.

You can change a book's page size and orientation at any time, even after you add objects to the pages. However, Instructor does not rearrange objects to fit the new page size or orientation. To fit the objects into a smaller page size or different orientation, move them to their new locations before you reduce or reorient the page. If necessary, you can return to the original page size to see the objects in their original positions.



Tip The smaller the page size, the faster pages appear and update.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To set a page size:

Using the interface

1 From the Object menu, do one of the following:

- To set a default page size for the entire book, choose Properties for Book.

  The Properties for Book dialog box appears.

- To set a custom page size for a background, choose Properties for Background.

  The Properties for Background dialog box appears.

2 Click the Page tab, and then enter the size you want in page units, pixels, or inches.

3 *Optional.* Close the Properties dialog box.



Tip You can change the default unit type from page units to either pixels or inches on the New tab of the Instructor Options dialog box. To open the Instructor Options dialog box, choose Options from the View menu.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The book's *size* property sets a default page size for the book. The background's *size* property can be set for individual backgrounds. This size affects only the pages that share that background.

- To set a default page size for the book or a background, set the *size* property. Page size is specified in Instructor page units ($^{1}/_{1440}$ of an inch):

```
size of this book = 10000,15000
size of this background = 10000,15000
```

- To view the page size of the book or the background, use one of the following commands:

```
put size of this book
put size of this background
request size of this book
```

  Instructor displays two numbers, representing the width and height of the page in page units.

Scaling the window size to the page size

The page size does not control the size of the window. A window can be scaled larger than a page (Instructor displays a mat beyond the page's edge) or smaller than a page (Instructor displays scroll bars). To scale the window to match the page size, use the Size to Page command on the View menu, or send the OpenScript *sizeToPage* message. Instructor expands or contracts the window to fit the current page or to the maximum window size, whichever is smaller.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Setting properties for books

You can set properties for a book just as you can set properties for all objects in Instructor. For example, the book's title, description, and passwords are all book properties. To set a book's properties, from the Object menu, choose Properties for Book. In the Properties for Book dialog box, make your selections.

Protecting books with passwords

To protect and control access to a book, you can assign separate passwords to allow access for opening, modifying, and saving the book. You can also assign the same password to all of these activities in one operation.



Important Be sure to write down your password. If you forget your password, you will not be able to perform password-protected activities.

➤ **To define a password:**

1 From the Object menu, choose Properties for Book.

2 On the General tab, click Set Passwords.

The Passwords dialog box appears.

3 Click one of the following buttons to open the Set Passwords dialog box:

- **Author** Specifies the password required for switching to Author level.
- **Open** Specifies the password required for opening the book.
- **Save** Specifies the password required for saving the book.
- **Set All** Specifies the password required for switching to Author level, opening a book, and saving a book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

4 In the Set Passwords dialog box, in the Password box, type the password required for the particular user activity, based on the button you clicked in step 3.

5 Retype the password in the Confirm password box, and then click OK.

6 In the Passwords dialog box, click OK.

7 *Optional.* Close the Properties for Book dialog box.



Note You can remove a password by clicking Clear Password or by deleting a password in the Set Passwords dialog box.

You can use the *passwordTypes* OpenScript function to return a list of the types of passwords that are active in the specified book. The *passwordTypes* function takes a <bookPath> parameter. This parameter must be a valid book path. For example:

```
CurrentPasswords = passwordTypes("c:\instruct\WebBook.tbk")
CurrentPasswords = passwordTypes(name of this book)
```

Navigating through a book

There are many ways to allow for navigation through a book. You can use Instructor navigation objects from the Catalog, you can assign navigation actions to objects using the Actions Editor, or you can rely on the built-in navigation tools at both Reader level and Author level. You can specify that certain pages be skipped from navigating, specify conditions for skipping to a particular page or Uniform Resource Locator (URL), and set a transition effect for moving from one page to another.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using built-in navigation

You can move from page to page using the menus, the page symbols in the status bar, or the keyboard.

Moving from page to page

To move to this page...	From the Go menu, choose...	At Author level, click on the toolbar...	Or press these keys...
Next	Next Page		ALT+RIGHT ARROW
Previous	Previous Page		ALT+LEFT ARROW
First	First Page		ALT+UP ARROW
Last	Last Page		ALT+DOWN ARROW

Using navigation objects

The Catalog contains navigation objects that you can use to help the user navigate through your application. These navigation objects feature graphics and hyperlinks that will jump to the next, previous, first, or last page in your application. The simplest use of these objects is to place one on the background of a book. The navigation object will appear on each page that shares that background. You can also create your own navigation buttons using the tool palette and set hyperlinks to move to another page.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Skipping pages when navigating

If your book contains pages that you do not want the user to see, you can prevent navigation to those pages at Reader level using the Properties for Page dialog box or OpenScript. This feature is especially useful if you're displaying pages of a book in a viewer and want to limit the user's access through the book. For details about viewers, see Chapter 19, "Creating windows with viewers."

➤ **To prevent navigation to a page at Reader level:**

Using the interface

1 Go to the page to which you do not want the user to navigate.

2 From the Object menu, choose Properties for Page.

3 On the Behavior tab, select Skip this page when navigating.

4 *Optional.* Close the Properties for Page dialog box.

Using OpenScript

◆ Set the *skipNavigation* property for the page to *true*:

```
skipNavigation of page ID 25 = true
```

Setting up transition effects between pages

You can set up transition effects when the user moves between pages. For example, you can create the effect that one page is sliding off to the left while another page slides in from the right. Transition effects are specified in the Hyperlink dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To specify a transition effect:**

1 Select the object or word that contains the hyperlink to another page (for example, the Next button in a navigation panel).

2 From the Object menu, choose Hyperlink.

3 In the Type of link list, select Go to another page.

4 Under Style, select a transition effect from the Transition effect list. Click Test to see a sample of the selected effect *(optional)*.

5 Click OK.

Adding navigation behavior to an object

Using the Actions Editor, you can program any Instructor object to send a user to another page, book, or URL, or to exit from a lesson. One of the benefits of using an action sequence to navigate is that you can specify a destination based on a specific condition or set of conditions. You can use the Actions Editor to quickly define action properties, such as destination page and transition effects, and then add the action to an object's action sequence. For more information about programming in the Actions Editor, see Chapter 15, "Using the Actions Editor," and Chapter 16, "Creating action sequences: Practical examples."



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To add a navigation action to an object's action sequence:

1. Select the object.
2. From the Object menu, choose Actions.

   The Actions Editor appears.
3. In the Event list, select the event that activates the action sequence.
4. From the Insert menu, point to Action, point to Navigation, and then do one of the following:
   - To specify navigation to any page in the current book or in another ToolBook II book, or to open a page in a pop-up viewer, choose Go to Page.
   - To specify exit from the lesson, choose Exit Lesson.
   - To display a document created in another application or to specify navigation to a particular URL, choose Display Document.
5. In the Properties for Action dialog box, select a destination, and then set any other action properties.
6. Click OK.
7. In the Actions Editor, from the File menu, choose Update Actions & Close.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Enhancing a book's performance

Careful page and background design can greatly enhance the performance of your book. When planning your book, use these guidelines:

- Limit the number of objects and graphics in a book.
- Make a book's page size as small as possible.
- Use the same background for similar pages and limit the number of different backgrounds.
- Put objects that appear on every page on the background.
- Arrange objects so that changes affect small areas to minimize redrawing.
- Do not overlap record fields, a type of field that contains text on backgrounds, and other objects. If they overlap, Instructor must redraw both the record field and its text when you change the page. Allow about 30 page units between objects to prevent overlapping.

Testing is an essential step in building a book. Test each part of your book as you complete it. Switch to Reader level to see how the pages will look to the user and try out buttons and hotwords to make sure they work. After the book is complete, test it as a whole. Go through it yourself to see whether the overall design and organization work. Then get someone else to try it—a fresh perspective can be invaluable.

For more information about enhancing the performance of your book, see Chapter 22, "Optimizing Instructor applications."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 5

Working with the Catalog

The Catalog contains a rich collection of objects—from simple navigation buttons to sophisticated objects such as the Universal Media Player and interactive questions. You can use the Catalog to find objects to use in your applications. Additionally, you can create your own objects and store them in the Catalog to use in other applications you create. This chapter describes how to use the ToolBook II Instructor Catalog.

IN THIS CHAPTER

About the Catalog **126**

Authoring with the Catalog **127**

Exploring types of Catalog objects **132**

Customizing the Catalog **137**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About the Catalog

The Catalog contains a collection of objects that you can use to construct and enhance interactive applications. Authoring is fast and easy using the objects in the Catalog because you can simply drag the object you want directly to your page. If you like, you can use the Catalog as your primary authoring resource since it includes virtually everything you need, from simple draw objects to sophisticated interactive objects. More likely, you'll use the Catalog as a resource for preprogrammed objects such as questions objects, navigation panels, media players, and so on.

The Catalog window contains buttons that display the names of categories of objects. You click a category button to view the objects available in that category.

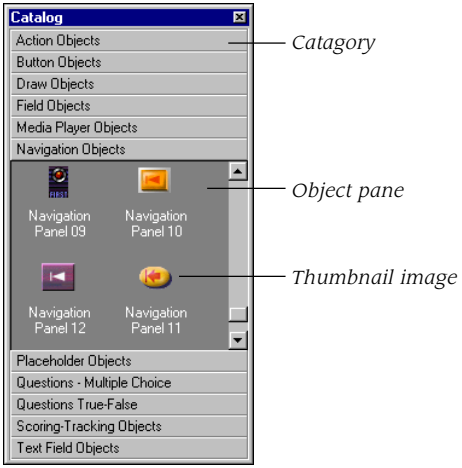


Figure 1
The ToolBook II Instructor Catalog

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To display the Catalog:**



◆ Click the Catalog button on the Instructor toolbar.

As with other windows, you can close, move, and resize the Catalog window.

Authoring with the Catalog

The Catalog is designed for ease of use. The intuitive browsing environment and drag-and-drop capabilities make it easy to build the pages of your book. You can add objects to the Catalog or remove them. The next few sections describe how to work with the Catalog.

Browsing through the Catalog

Take a moment to browse through the objects in the Catalog. When you select a category, small images of objects appear in the object pane below the category button you selected in the Catalog. You can display a description of any Catalog object: point to an object, right-click, and select Object Description from the right-click menu.

Using drag-and-drop authoring

To author using the Catalog, simply drag the object you want from the Catalog to a page or background. This drag-and-drop operation copies the object into your application.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

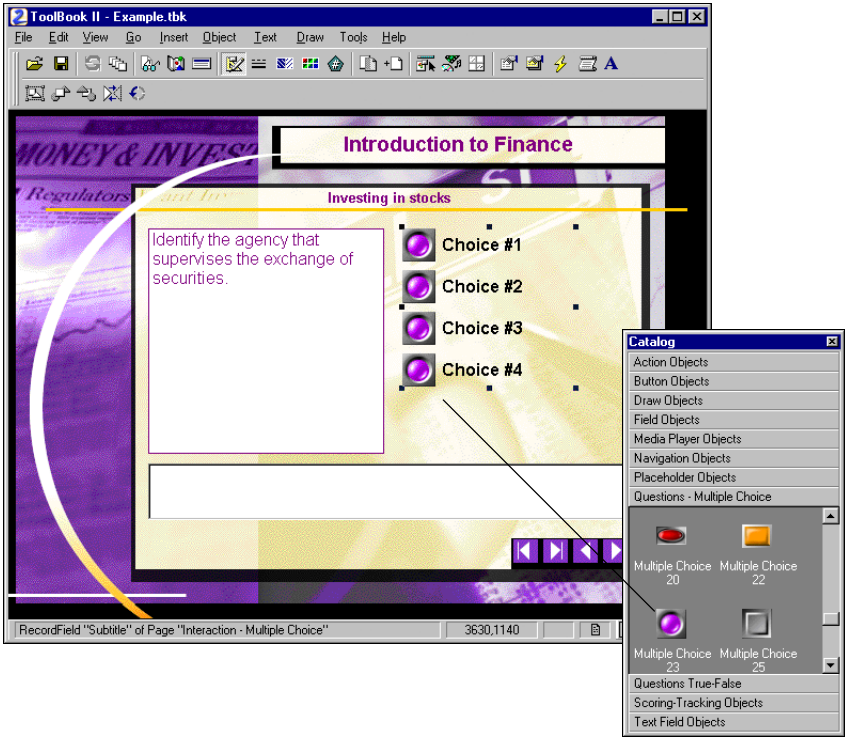


Figure 2
An object dragged from the Catalog to a page

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Some Catalog objects can be dropped on objects other than a page or background.

Drop behavior for Catalog objects

Dropping this...	On this...	Does this...
Graphic	Page or background	Creates a paint object. The Graphic Options dialog box opens, in which you can specify how you want to apply the graphic. You can apply the graphic as a bitmap on the page, or you can apply it as a tiled backdrop.
	Button	Creates a button graphic. The Button Graphic dialog box opens, in which you can specify how you want to apply the graphic.
	Field	Creates an inline graphic (a graphic set within text) at the pointer position.
Question object	Page or background	Creates a question object at the pointer position. The question's Extended Properties dialog box opens (optionally).
Object with extended properties	Page or background	Creates an object at the pointer position. The object's Extended Properties dialog box opens (optionally).

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

When an object in the Catalog has extended properties, Instructor can automatically open its Extended Properties dialog box after you drag the object to a page or background. You might find this feature convenient when you are first learning to use Instructor. To enable this feature, choose Options from the View menu; then, on the Interface tab in the Instructor Options dialog box, select Show editor after adding objects, and then click OK.

Finding the right object

You can identify an object you want in the Catalog by reading a description. Point at an object in the Catalog window, right-click and select Object Description from the right-click menu. You can also search on keywords to locate specific objects.

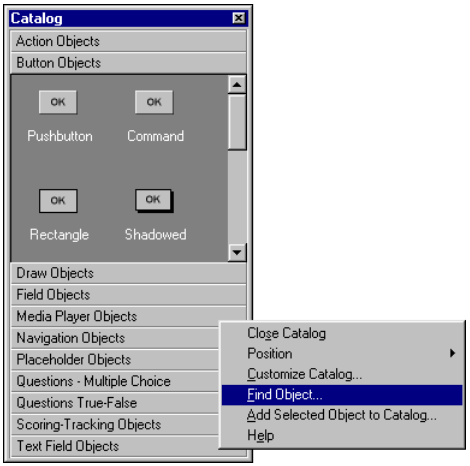


Figure 3
Right-click menu for categories in the Catalog

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To search for an object in the Catalog:**

1 Point in the Catalog window and right-click.

2 From the right-click menu, choose Find Object.

The Find Object dialog box appears.

3 Type a keyword in the edit box.

4 Click OK.

Instructor searches the Catalog for an item with the keyword you entered and displays the result.

Viewing objects

You might want to preview an object to view it at full size.

➤ **To view an object at full size:**

1 In the Catalog, select the small thumbnail image of the object you want to view.

2 Right-click the object. From the right-click menu, select Preview Object.

Instructor shows you a full-size view of the object in a pop-up window.

3 Click anywhere on the screen to close the pop-up window.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Exploring types of Catalog objects

The following sections give you a general overview of the objects available in the Catalog. By browsing through these sections, you can get an idea of the purpose of different types of objects.



For more information about individual objects in the Catalog, add the object to a page and read the contents of the ToolBook II Coach (press CTRL+F1) for specific information; alternatively, refer to online Help for more general information.

Action triggers

Action trigger buttons can start and reset objects on the page. With trigger and reset buttons, you can hide and show objects, play an animation, and create dynamic effects. You can trigger an action interactively by setting the effect to occur when the user clicks the action trigger at Reader level. Alternatively, you can trigger an action automatically by setting the effect to occur when the user enters the page. Among the action buttons you'll find are:

- **Action Trigger** A button that can show, hide, animate, and trigger another object.
- **Reset Trigger** A button that can reset another object.
- **Delay** A button that delays the next action for a specified amount of time.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Scoring and tracking

Evaluation buttons perform common tasks in courseware applications, such as submitting a lesson score to a learning management system, restarting a lesson, checking the learner's response, or resetting a page.

You use these objects for specific purposes on a page in a lesson, so there is no need to set properties for them. Examples of scoring buttons include:

- **Score Page** A button that displays the score for all of the question objects on the page.
- **Reset Page** A button that resets the questions on the current page and clears any score that the student has accumulated for the page. Session information is preserved and a log entry is made indicating that the page was reset.
- **Score Quiz** A button that displays the final score for a quiz or lesson.
- **Score Field** A text field that you can use in conjunction with a question object and a score button to display a score.
- **Feedback Field** A text field that you can use in conjunction with a question object and (optionally) a score button to display feedback.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Placeholders

The placeholders included with Instructor are objects that you can use to display images, embed HTML code, and integrate other Web technologies. You apply settings to each object using that object's Extended Properties dialog box. This category of objects is available only in books that will be exported to DHTML. Placeholders include:

- **HTML Placeholder** A placeholder in which you can place any HTML code that you want to include in an application that will be exported for the Web.
- **File Placeholder** A placeholder in which you can place any object that can be inserted into a Web page using an HTML embed tag. If your users' machines are capable of running the file (that is, they have the appropriate plug-in or helper application installed), the file will run.
- **VRML Placeholder** A placeholder that allows users to run Virtual Reality Modeling Language (VRML) files that you specify.
- **Java Placeholder** A placeholder that includes a Java applet in an application that will be exported for the Web.
- **Animated GIF Placeholder** A placeholder that allows users to view GIF (Graphics Interchange Format) animations that you specify.
- **PowerPoint Placeholder** A placeholder that allows users to view a PowerPoint® presentation in a Web browser.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Media players

Media players are objects that you add to your application to enable users to play sound, video, or animation files. The Universal Media Player is the primary object for adding media files to an Instructor application: It supports most standard media types, is programmable using the Actions Editor, and retains all functionality when exported to DHTML. The Catalog also includes several other media objects with more specific functionality. Among the objects you'll find are:

- **Media Player** A button that plays a specific media type.
- **Media Player (hotword)** A hotword in a text field that plays a media file.
- **Video Player** A simple stage or a stage with controls that plays visual media.

Navigation

Navigation objects are buttons and fields that enable students to move through your application and that can also display the current location. Some examples include:

- **Automatic Menu** An object that automatically creates a menu that includes hotwords that jump to each named page in your book.
- **Button Group** A grouped object consisting of buttons for general navigation.
- **Hotspot Graphic** An object that produces an image with areas that can be clicked. You can define hyperlinks for the hotspots in the graphic, which will cause navigation to a different page in a Web browser.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Questions

Question objects allow you to create questions that test your users on their understanding of your content. You can customize question objects in many ways. These objects have built-in behavior that provides feedback to your users, supplies you with information on how your users interacted with the question, and interacts with a learning management system to track the behavior of groups of learners. Among the question objects provided with Instructor are the following:

- **Match-items object** A button set that allows the user to drag an arrow to define pairs of matching items.
- **Multiple-choice object** A button set that allows the user to select the correct answer by clicking one of several available choices.
- **Fill-in-the-blank object** A field that allows the user to enter text for an answer and then evaluates whether the text is correct.

Other available objects

The objects mentioned in the preceding sections are just a few of the multitude of objects that are available in the Catalog. Perhaps the best way to learn what is available in the Catalog is to explore them as you work. Soon you will become an expert at locating the objects you need when you need them.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Customizing the Catalog

You can change many aspects of the Catalog to fit your personal preferences. For example, you can modify the display of the categories in the Catalog so that you see only the categories that you plan to use in your current book. In addition, you can add new categories to the Catalog and reorder the existing categories. You can add your own customized objects to the Catalog and import customized Catalog objects created by other ToolBook II authors.

Modifying the categories in the Catalog

You can change the contents of any category that you see in the Catalog, and modify the properties for the Catalog object icon images displayed in the Catalog window. In the Customize Catalog dialog box, you can edit the existing categories and add new categories to the Catalog.

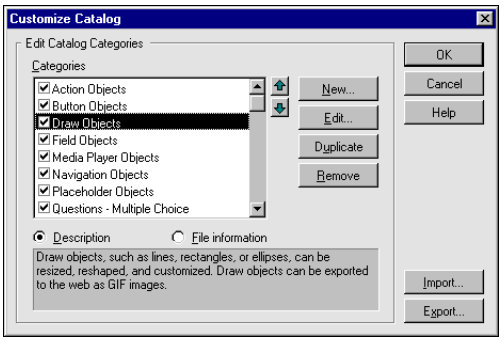


Figure 4
The Customize Catalog dialog box

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



Tip In the Customize Catalog dialog box, you can clear the check box next to a category to hide the selected category from view. This does not permanently delete the category from the Catalog.

Adding and removing Catalog objects

You can add your own objects to the Catalog. For example, if you have customized the behavior of an object using the Actions Editor visual programming tool, you may want to store that object in the Catalog to use again. You can add your own customized object to the My Objects category in the Catalog, or to any of the existing Catalog categories. You can also create a new category for the Catalog.

When you add an object to the Catalog, you can assign it a name and search keywords. If you want to enter multiple search keywords, use commas to separate them.

➤ **To add an object to the Catalog:**

1 In the main window, select the object you want to add to the Catalog.

2 Open the Catalog if it is not already open.

3 Right-click on a category in the Catalog window.

4 From the right-click menu, choose Add Selected Object.

The Add Selected Object dialog box opens.

5 Type a name for the object, a description if desired, and any search keywords. Select or create a category, and then click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To remove an object from the Catalog:

1. In the Catalog, select the object you want to delete.
2. Press the DELETE key.

   Instructor asks whether you want to remove the current object from the Catalog.
3. Click OK to remove the current object. Click Cancel to retain the current object in the Catalog.

Instructor removes the Catalog object from view.

➤ To remove a Catalog object from view:

1. Right-click on a category button in the Catalog.
2. From the right-click menu, choose Customize Catalog. The Customize Catalog dialog box appears.
3. Select the category that contains the object you wish to remove.
4. Click the Edit button.
5. Choose the object you with to remove from the object list.
6. Click the Remove button.

Instructor removes the Catalog object from view, but does not delete the object from the Catalog.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 6

Working with objects

Circles, squares, lines, graphics, buttons, and questions are all examples of ToolBook II Instructor objects. You can use these objects to create an application with your own unique design. This chapter introduces you to the concepts, skills, and tools you need to add and modify objects in Instructor.

IN THIS CHAPTER

About objects **142**

Adding objects from the Catalog **142**

Creating objects using the tool palette **144**

Selecting objects **147**

Creating hyperlinks **148**

Cutting, copying, pasting, duplicating, and deleting objects **151**

Moving objects **153**

Spreading and centering objects **154**

Flipping and rotating objects **154**

Aligning objects **155**

Changing the size and shape of objects **157**

Modifying the appearance of objects **160**

Moving objects between layers **166**

Grouping objects **172**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About objects

All the visual elements of your application—buttons, fields, graphics, viewers, and even the pages and backgrounds—are objects. You build your application by creating objects, specifying their properties, and defining how they will work together. ToolBook II Instructor offers a variety of objects with built-in functions for quick and easy authoring. But you aren't limited to these preprogrammed features: You can position, group, or change the properties of any object to affect its appearance and behavior.

When an object is created or otherwise added to a book, Instructor gives the object a unique ID number that you can use to distinguish it from other objects. You can name the object, but you cannot change its ID number.



Note It is a good idea to assign each object in your book a unique name. Unnamed objects will not appear in lists of available objects in the Actions Editor or in the Extended Properties dialog box for some objects.

Adding objects from the Catalog

You can quickly add objects to a page or background. At Author level, you simply drag the object you want from the Catalog to your page. This creates a new instance (or copy) of the object on your page.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

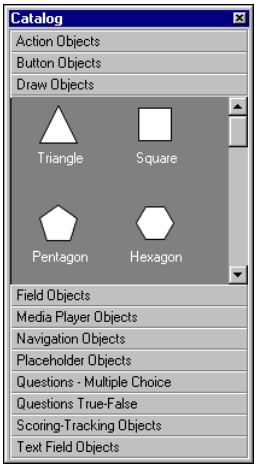


Figure 1
Draw objects in the Catalog

➤ **To create an object on a page or background:**

1 Navigate to the page or background to which you want to add an object.



2 If a the Catalog is not already open, click the Catalog button on the Instructor toolbar.

The Catalog appears.

3 Select a category.

4 Drag the object you want from the object pane, and then drop it onto your page or background.

Instructor draws an instance (copy) of the object on the page or background.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Tip You can double-click an object in the Catalog to add it to your page.

After you add objects to a page or background, you can customize them for your application. Objects have built-in properties that define how they appear and behave. For example, you can set properties such as fill color, animation, and hyperlinks. Information about setting properties is available in Chapter 7, "Setting object properties."

Creating objects using the tool palette

In addition to adding objects to your book by dragging them from the Catalog, you can draw objects using tools from the tool palette. While Catalog objects have a predefined appearance and certain predefined behaviors, objects created using the tool palette have limited preprogrammed properties. You can use the Actions Editor visual programming tool to easily link an action or series of actions to any object you create using the tool palette. Alternatively, you can use the OpenScript programming language to further define an object's properties and behavior.

When you draw an object, its general shape and type are determined by the tool you choose from the tool palette, and its final size and proportions are determined by how far and in which direction you drag the pointer. When you select an object or use a tool, the pointer shows the type of object selected.

The tool palette, shown in Figure 2, is available only at Author level and appears by default on the left side of the main window. However, you can reshape the tool palette and move it to a new location to suit your work style.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

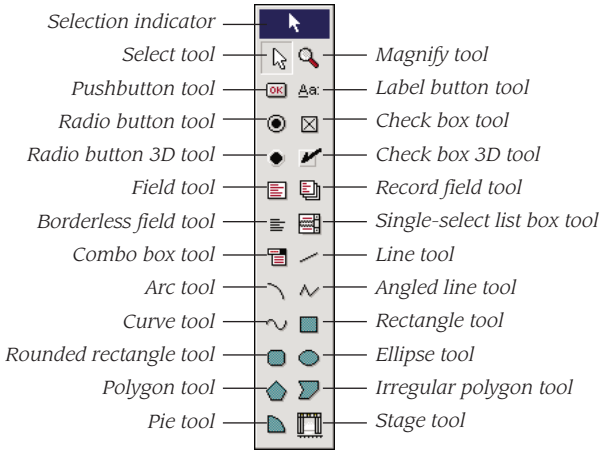


Figure 2
Instructor tool palette

➤ To draw an object:

1 Choose a tool from the tool palette by clicking the tool's button.

2 Point where you want to start the new object.

3 Hold down the left mouse button while you drag, and then release when the object is the size you want.

You use a slightly different method to draw polygons, angled lines, irregular polygons, and curves.

Tip To automatically select the object you've just drawn, press the SPACEBAR. Instructor selects the object and displays the select tool.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To draw regular polygons:



1. Choose the polygon tool from the tool palette.
2. From the View menu, point to Palettes, and then choose Polygon.
3. From the polygon palette, choose one of the polygon shapes, or type a number in the box to specify the number of sides for the polygon.
4. Hold down the left mouse button where you want the center of the polygon, and then drag from the center and release when the shape is the size you want.

➤ To draw angled lines and irregular polygons:



1. Choose the angled line or irregular polygon tool from the tool palette.
2. Point where you want the shape to start, and then click.
3. Continue to point and click to add as many angles as you want.

   Press CTRL if you want to constrain the line to 45-degree angles.
4. To end the shape, double-click or press ESC.

   If you used the irregular polygon tool, Instructor draws a closed shape filled with the currently selected pattern from the pattern palette.



Note You can use the angled line tool to draw an object that looks like an enclosed object. However, avoid having users click these objects, because they require clicking precisely on the line. If you want an irregularly shaped enclosed object, create the shape with the irregular polygon tool.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To draw curves:**



1 Choose the curve tool from the tool palette.

2 Point where you want the curve to start, and then click.

3 Point where you want the curve to begin curving, and then click.

4 Continue to point and click to add as many curves as you want.

5 To end the curve, double-click or press ESC.

After you draw a curve, you can adjust it precisely by choosing the Reshape Curve command from the Draw menu. For details, see "Changing the size and shape of objects," later in this chapter.

Selecting objects

Before you can change an object or define its properties, you must first select the object. When you select an object, Instructor displays handles around the object. You use the handles to align and size objects.

You can select single or multiple objects. When you select multiple objects, your next command or mouse action affects all of the selected objects. For example, you can drag them, change their properties, and cut or copy them as if you had selected a single object.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Selecting objects

To...	Do this...
Select a single object	Click the object.
Select multiple objects	SHIFT+click each object you want to select, or drag the pointer to select the objects you want.
Select all the objects on the page or the background	Go to a page or background and choose Select All from the Edit menu, or press CTRL+A.
Cancel a selection	Click anywhere on the page away from the object.
Add or remove objects from a selection	SHIFT+click the object you want to add or remove.

Creating hyperlinks

Hyperlinks allow the users of your application to move from one page to another. A user can display a page out of sequence or even display a page in a different application. All objects you can put on a page have built-in hyperlink capabilities that let you connect ideas, information, or topics.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

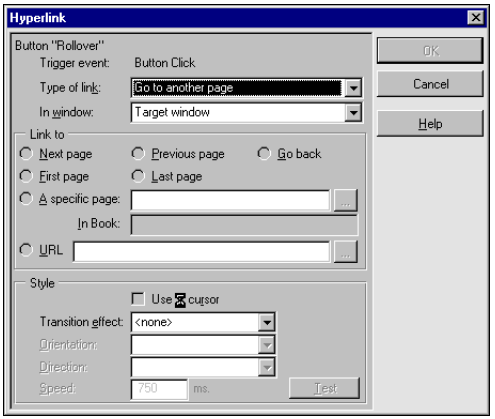


Figure 3
The Hyperlink dialog box

Choosing a type of link

You have several options when choosing the type of link you want. You can define a hyperlink that navigates to another page or one that pops up a page. For example, you can add a button to a page that, when clicked, takes the user to another page or opens a pop-up window. When you create a hyperlink to another page, you can also specify a transition effect.

A hyperlink that pops up a page is useful when presenting technical content to both novice and expert users. For example, you can create hotwords for the technical terms in your content that, when clicked, open a pop-up window containing the definition of a term. When you create a hyperlink that pops up a page, you can specify the style of the pop-up window.

In addition, the destination of a hyperlink can be either a page in a book or a URL on the Internet. When creating a hyperlink to a URL on the Internet, you can define objects that launch a Web browser and jump to a Web site, as described in the table on the following page.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Hyperlinking to a URL

Choose this option...	If you want to define a hyperlink that...
Go to another page	Jumps to the specified URL in the current window of the Web browser. If the Web browser is not open at the time of the jump, Instructor will open it before jumping to the specified URL.
Pop up another page	Opens a new instance of the Web browser before jumping to the specified URL.

➤ To create a hyperlink:

1. Select the object that you want to hyperlink.
2. From the Object menu, choose Hyperlink.

   The Hyperlink dialog box appears.
3. In the Type of link list, select the kind of link you want to use.
4. Under Link to, select the type of link you want.
   - To link to another page, select the page option you want.
   - To link to a URL, enter a URL in the URL box.
5. *Optional*. Specify a transition effect for the hyperlink. Transition effects are available only when you choose Go to another page as the type of link you want to use.
6. *Optional*. Under Pop up style, select a window style in the list. Pop up style is available only when you choose Pop up another page as the type of link you want to use.
7. Click OK.

For more information about hyperlinks, refer to online Help.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Cutting, copying, pasting, duplicating, and deleting objects

You can share an object among pages by cutting, copying, pasting, and duplicating it. Cutting removes an object from a page and places it on the Clipboard so that you can paste it either in the current book or in another book. When you copy an object, Instructor places a copy on the Clipboard, so you can then paste it. When you duplicate an object, you create an exact copy of the object on the same page, bypassing the Clipboard.

Most objects can be duplicated. For example, you can duplicate an entire text field or record field. The exception is text: You cannot duplicate selected text within a text or record field. You can cut or copy and paste objects, as described in the following table.

Objects that can be cut, copied, and pasted

This object...	Can be pasted...
Object	On a page or background
Page with background	After the current page
Page without background	After the current page
Bitmap	On a page or background, or in a text field or record field
Picture object	On a page or background, or in a text field or record field
Text	In a text field or record field
Text in Rich Text Format	In a text field or record field

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To cut or copy an object:

1 Select the object you want to cut or copy.

2 From the Edit menu, choose Cut or Copy.

- To remove the object from its current position, choose Cut.
- To keep the object in its original position and place a copy elsewhere, choose Copy.

Instructor places the cut or copied object on the Clipboard.



Tip With an object selected, you can also use the standard Windows shortcut keys CTRL+X, CTRL+C, and CTRL+V to cut, copy, and paste.

After you cut or copy an object, you can paste it on the same page or anywhere else in the current book or in another book.

➤ To paste an object:

1 Go to the location where you want to paste the object.

2 From the Edit menu, choose Paste.

Instructor pastes the object on the current page in the same position the object held on the original page. If you choose the Paste command again, a second object is pasted on top of the first. To reveal the first object, select the top object and drag it to a new position.

➤ To duplicate an object:

1 Select the object you want to duplicate.

2 From the Edit menu, choose Duplicate.

Instructor creates a duplicate and places it slightly below and to the right of the original object.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can remove an object from any page by deleting it.

➤ **To delete an object:**

1 Select the object you want to delete.

2 From the Edit menu, choose Delete, or press the DELETE key.

Instructor removes the selected object.

If you delete an object accidentally, immediately choose Undo from the Edit menu to restore it; otherwise, the object will be lost.

Moving objects

You can move an object to another location on a page by dragging it to a new location or by "nudging" it pixel by pixel.

➤ **To move an object to another location on a page:**

◆ Do one or a combination of the following:

— Drag the object to its new location.

— Nudge the object by clicking it and using the arrow keys to move the object one pixel at a time in the desired direction. Press SHIFT as you're using the arrow keys to move the object faster.



Tip When dragging an object, press the CTRL key if you want to restrict the object to horizontal or vertical movement.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Spreading and centering objects

You can use the Spread and Center commands on the Draw menu to automatically spread or center a selection of objects on a page or background. The Spread command distributes a set of selected objects along a horizontal or vertical axis on the page. The Center command centers a set of selected objects between the left and right sides of a page, top and bottom of a page, or both.

➤ **To spread a selection of objects:**

1 Select the objects you want to distribute.

2 From the Draw menu, point to Spread, and then choose an option from the submenu.

➤ **To center a selection of objects:**

1 Select the objects you want to center.

2 From the Draw menu, point to Center, and then choose an option from the submenu.

Flipping and rotating objects

You can transform Instructor draw objects by flipping or rotating them. Flipping transposes an object on an axis. Rotating moves an object around a center point. You can combine flipping and rotating to move an object to the position you want.



Note You cannot flip or rotate some objects, such as imported graphics, button captions, and text in a text field.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To flip or rotate an object:**

1 Select the object or objects you want to transform.

2 From the Draw menu, choose one of the following commands:



- **Flip Horizontal** Flips the selected object horizontally within its bounding box (the rectangular area that indicates the object's dimensions), switching the location of the object's left and right sides.



- **Flip Vertical** Flips the selected object vertically within its bounding box, switching the location of the object's top and bottom.



- **Rotate Left** Rotates the selected object 90 degrees to the left (counterclockwise).



- **Rotate Right** Rotates the selected object 90 degrees to the right (clockwise).

Aligning objects

You can use the following methods to precisely position objects on a page:

- Arrange objects using the grid as a reference
- Snap objects to the nearest grid point for precise placement
- Measure and align objects with the rulers
- Align objects with one another using the Align command

For information on using the grid and rulers, see Chapter 3, "Learning the Instructor interface."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the Align command

When you create diagrams and other complex objects, it may be important to align the objects. For example, you might need to align rows of fields along their top edges, or to center columns of fields. Use the Align command on the Draw menu to align several selected objects in precise relation to each other.

When you select multiple objects, Instructor uses one object as the reference against which it aligns the others. For example, to move objects so that their left edges align with the left edge of the leftmost object, from the Draw menu, point to Align, and then choose Left.

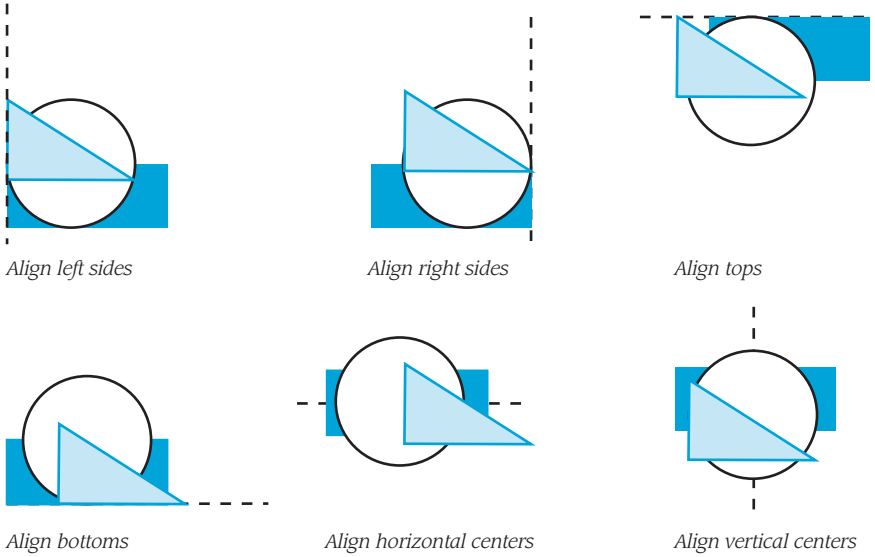


Figure 4
Aligning objects with each other

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To align objects with each other:**

1 SHIFT+click to select the objects you want to align.

2 From the Draw menu, point to Align, and then choose an option from the submenu.



Note The Align command aligns objects by their bounding box, which can be seen when the object is selected. The bounding box of a curve, arc, or pie wedge can extend past the visual portion of the object; it may be easier to align these types of objects manually.

Changing the size and shape of objects

You can change the size and shape of any object manually. You can also use the Size command on the Draw menu to automatically size one object to another.

➤ **To change the size of an object:**

1 Select the object you want to resize.

2 Drag a handle until the object is the size you want.

- To change an object's width and height at the same time, drag a corner handle.
- To change an object's width, drag a side handle.
- To change an object's height, drag a top or bottom handle.
- To resize an object proportionally, press CTRL as you drag a corner handle.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

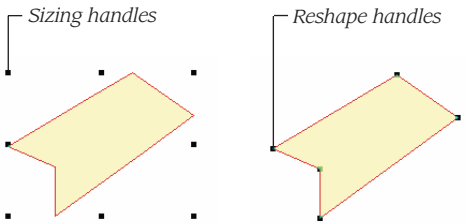
If you drag the handle of one object when several objects are selected, only the object with the handle you drag resizes; the other objects maintain their original sizes. To resize several objects at once, you must group them. For details about grouping objects, see "Grouping objects," later in this chapter.

To change the appearance of a polygon, curve, arc, angled line, or pie wedge, you can use the Reshape command to display the object's reshape handles, which are handles that can be used to alter the shape of these objects.

➤ **To reshape an object:**

1 Select the object (angled line, curve, arc, polygon, or pie wedge) whose shape you want to change, and then from the Draw menu, choose Reshape. Alternatively, double-click the object.

The handles change to allow you to reshape the object.



2 Drag the handles to change the shape of the object.

For arcs and pie wedges, you can drag the handles only along the curvature of the shape. For other objects, you can drag the handles in any direction.

3 To remove the reshape handles and restore the sizing handles, click the object, or click elsewhere on the page to cancel the selection.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

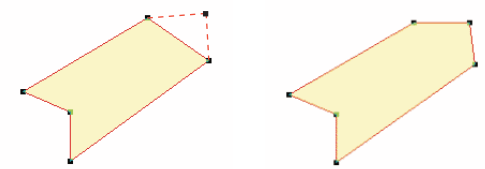
Glossary

Index

For more precise control over the shape of some objects, you can add reshape handles that enable you to add another vertex to a side of the object.

➤ **To add reshape handles to an object:**

1. Select the object (angled line, curve, or regular or irregular polygon), and then choose Reshape from the Draw menu. Alternatively, double-click the object.
2. Press SHIFT, and then click an existing handle. A new handle appears that you can drag to add another vertex to the object.



➤ **To delete a reshape handle:**

- Press CTRL+SHIFT while clicking a reshape handle.

If the object is a curve, the handles are removed in pairs. You cannot remove the last two handles in an object. You cannot remove the last three handles of a polygon.

Constraining an object's shape

With Instructor, you can proportionally constrain objects or constrain lines to certain angles. To constrain an object's shape, press CTRL while dragging. The table below shows how objects are constrained when you press CTRL.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Constraining objects

Press CTRL with this tool...	To constrain...
	Rectangle to square, or rounded rectangle to rounded square
	Ellipse to circle
	Line angles to multiples of 45° (0°, 45°, 90°, and so on)
	Rotation to multiples of 45° (0°, 45°, 90°, and so on)
	Shape to quarter of a circle



Note You can also press CTRL before you move an object to restrict its movement to horizontal and vertical directions, which is useful when you want to move objects that are precisely aligned with one another.

Modifying the appearance of objects

You can modify the appearance of an object by changing its line style, fill color, and fill pattern. You can also specify that an object or background take its color settings from a user's operating-system settings. This allows you to create applications with a color scheme that will appear as a natural part of a user's Windows interface. You can also make an object transparent or opaque.

This section describes how to change the appearance of objects by setting properties. For additional information about setting properties, see Chapter 7, "Setting object properties."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Applying line styles

You specify the style of a line—such as solid, dashed, or dotted—the width of a line, and the type of line ends in the Properties dialog box. You can also change the line style of an object's bounding box. A bounding box is the rectangular area around a graphic object, such as a bitmap or paint object, that indicates its dimensions.

➤ To specify line style or width:

1 Do one of the following:

- To modify a line, select the line you want to modify.
- To modify an object's bounding box, select a graphic object.



2 Click the Properties button on the Instructor toolbar.

3 Click the Draw tab, and then choose the line style you want in the Line style list.

4 *Optional*. Close the Properties dialog box.

➤ To specify line ends:

1 Select the line you want to modify.



2 Click the Properties button on the Instructor toolbar.

3 Click the Draw tab, select an end in the End1 list to appear as the line's starting point, and then select an end in the End2 list to appear at the line's ending point.

4 Select a value for the size of each line end (from 1 to 9).

5 *Optional*. Close the Properties dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Applying color to objects

Fill color is the color of the inside of all draw objects. Stroke color is the color of the outline of any draw object, text, or imported object. Enclosed objects, such as rectangles and ellipses, have both a stroke (outline) and a fill (interior) color. You apply colors and patterns to an object in the Properties dialog box.

➤ **To apply color to an object:**

1 Select the object you want to color.



2 Click the Properties button on the Instructor toolbar, and then click the Draw tab.

3 Under Colors, choose Use custom colors, and then do one of the following:

- To assign a fill color to the object, click the Fill color button.
- To assign a stroke color to the object, click the Stroke color button.

A pop-up window appears, where you can choose from a palette of commonly used colors.

4 *Optional.* To create a new custom color, click Other, and then do the following:

A Click Define Custom Colors, and then click an empty Custom colors box on the left side of the dialog box.

B Click the color in the color matrix that is close to the color you want, and then use the slider to the right of the color matrix until the color you want appears in the Color/Solid box.

C Click Add to Custom Colors.

D Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

5 *Optional*. Close the Properties dialog box.

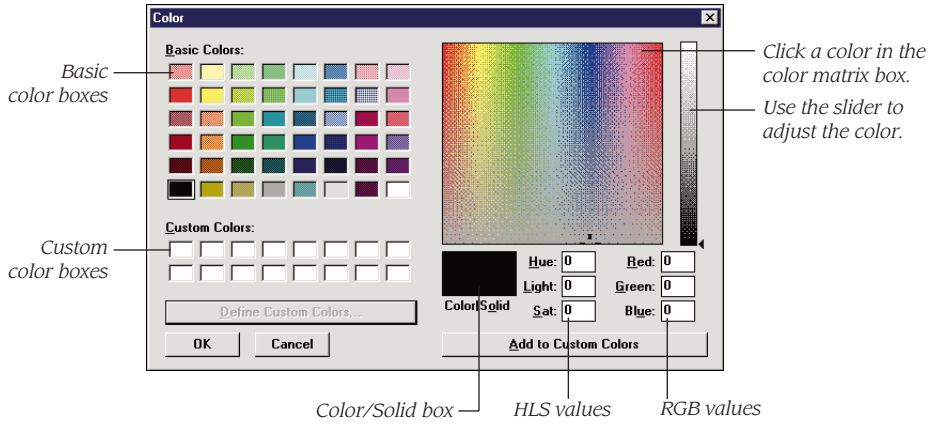


Figure 5
The Color dialog box

Using Windows colors

You can set the Use Windows Colors command on the Draw menu to specify whether an object or the background uses Windows system colors. If you specify that an object use the Windows system colors, the object's color will be determined by the user's system color settings.

Choosing this command does not change the fill and stroke colors that you have set for the object; it just overrides those settings. If you clear the Use Windows Colors command, Instructor will revert to using the object's fill and stroke colors.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To use the Windows system colors for an object:**

1 Select the object or display a page that shares the background that will use the Windows colors.

2 From the Draw menu, choose Use Windows Colors.



Tip Instructor can also draw a background using the dialog box color that is part of an operating system's color scheme. To draw a background using a dialog box color, on the Draw tab of the Properties for Background dialog box, choose Use standard Windows colors, and then select Use dialog box color.

Making objects transparent

You can create transparent objects or make existing objects transparent. When an object is transparent, objects on layers below it show through.

A transparent object's stroke and fill colors are also transparent. For example, if the transparent object is blue and an object on a lower layer is green, any overlapping areas will appear as a third color that Windows blends. You can use transparent objects to overlay text on objects, such as titles without a box around them.

➤ **To make an object transparent:**

1 Select the object you want to make transparent.

2 From the Draw menu, choose Transparent.



Tip To make an object invisible, set its fill and stroke colors to white, and then make it transparent. If you lose an invisible object on the screen, choose Select All from the Edit menu. Instructor displays handles for all objects on the page.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Filling an object with a pattern

When you draw enclosed shapes such as rectangles, polygons, and pie wedges, you can give them a look of greater depth by filling them with a pattern. Use one of the 128 Instructor patterns to fill your objects with bricks, stripes, or other abstract shapes. Instructor draws the pattern using the current stroke color. You will achieve a subtle effect if the stroke color is close in hue to the fill color. For greater contrast, choose a stroke color that is much darker or lighter than the fill color.

➤ **To fill an object with a pattern:**

1 Select the object.



2 Click the Properties button on the Instructor toolbar.

3 On the Draw tab, in the Pattern list, select a pattern.

4 *Optional*. Close the Properties dialog box.



Note You cannot apply a pattern to fields, buttons, or video players.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Moving objects between layers

When running an application at Reader level, a user sees a unified page in the main window. However, each page can consist of many layers. A layer is where Instructor places an object in the relative order of all objects on either the page or background. You do not see a layer; you see only the objects on a layer.

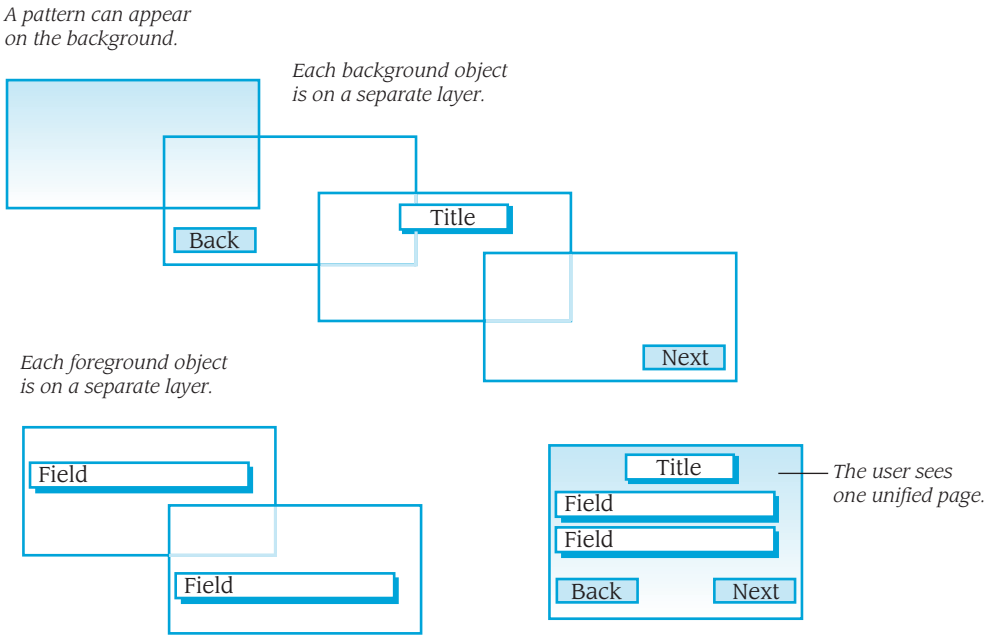


Figure 6
The layers of a page

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

When you create a new object, Instructor adds a layer to the front of the current page or background and then places the new object on that layer. The layer order affects the following aspects of a page:

- When objects overlap, the objects on higher layers appear to be placed on top of objects on lower layers.
- When tabbing between buttons and fields, the focus moves in order from the object on the lowest layer to the object on the highest layer. For more information about the focus and tabbing, see "Planning layer order for tabbing at Reader level," later in this chapter.
- When Instructor searches for text, it starts searching from the field or button with the focus and then searches fields on higher layers before continuing to the next page. Any fields on layers lower than the object with the focus will not be searched until after Instructor searches the rest of the book.

Changing layer order

When you change an object's layer number, Instructor assigns new layer numbers to all other objects affected by the change. For example, on a five-layer page, moving the layer 3 object to layer 4 puts the layer 4 object on layer 3. You can change an object's layer order by typing a different layer number in the Properties dialog box for the object. The total number of layers for the current page or background is shown beside the Layer box in an object's Properties dialog box. You can also choose one of the four layer commands from the Draw menu to change an object's layer number.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To move an object to the next higher or lower layer:**

1 Select the object.

2 From the Draw menu, point to Layer, and then choose Bring Closer or Send Farther.

Bring Closer moves the object one layer up, or forward, in the layer order. Send Farther moves the object one layer down, or back, in the layer order.

Select an object...



...and then choose Bring Closer to move the object up one layer.

Figure 7
Changing layer order

➤ **To move an object from its current layer to the back layer:**

1 Select the object.

2 From the Draw menu, point to Layer, and then choose Send to Back.

➤ **To bring an object from its current layer to the top layer:**

1 Select the object.

2 From the Draw menu, point to Layer, and then choose Bring to Front.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To specify an object's layer number:**



1 Select the object, and then click the Properties button on the Instructor toolbar.

2 Type a number in the Layer box.

3 *Optional*. Close the Properties dialog box.

Planning layer order for tabbing at Reader level

A user can press the TAB key to move the focus between objects on a page. When an object has the focus, it can receive input from the user. For example, when a button has the focus, its caption is highlighted; when a text field has the focus, a cursor appears in the field.

The tab order—the order in which the focus moves when a user presses TAB—follows the layer order of the objects, beginning with layers on the background and then moving to layers on the page. If the layer order is the order in which you created those objects, the tab order could seem random to the user. For example, the first field you create may be at the bottom of the page, the second at the top, and the last in the middle; your user has to look all over the page for the focus. To make your application easier to use, you can define a new tab order.



Note The tab order you set is not always maintained when you export your application to DHTML and view it in a browser.

There are two ways to determine the order in which readers will tab through objects on a page: by setting tab order in the Set Tab Order dialog box and by changing the layer number of objects in the Properties dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set the tab order using the Set Tab Order dialog box:**

1 Select all the objects for which you want to define the tab order.

2 From the Draw menu, point to Layer, and then choose Set Tab Order.

3 In the Set Tab Order dialog box, specify the starting layer and whether the order will follow from left to right or top to bottom.

4 Click OK.

You can also define a tab order by specifying the layer for each object in the Properties dialog box.

➤ **To set the tab order by changing layer numbers in the Properties dialog box:**



1 Select the object that is to receive the focus first, and then click the Properties button on the Instructor toolbar.

2 In the Layer box, type a number that is smaller than the layer number of any other fields or buttons to which users will tab.

3 Repeat this procedure for the remaining objects in the order you want tabbing to occur, giving each object an incrementally higher layer number. Then, switch to Reader level and test the tab order by pressing TAB and observing where the focus moves.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

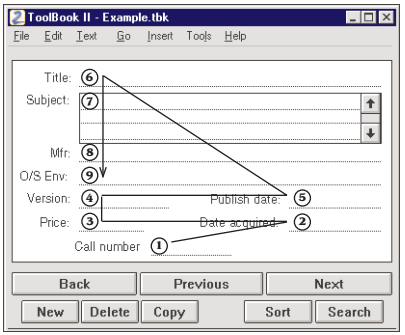
PART IV
Adding behavior with action sequences

PART V
Using advanced features

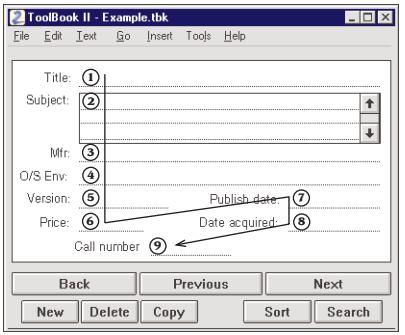
PART VI
Distributing and printing your application

Glossary

Index



Tab order initially follows the order in which fields were created.



Change the layer order of fields so that tab order is logical.

Figure 8
Planning tab order

Consider the following when you define the tab order:

- Try to follow a left-to-right and top-to-bottom reading pattern to organize fields and other objects on a page.
- If too many fields and buttons appear on a page, a user might have trouble finding the focus when pressing TAB. If your pages are crowded, consider simplifying them and adding new ones.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You may also exclude an object from the tab order:

- To exclude a button from the tab order, clear the Include button in tab order option on the Behavior tab of the Properties for Button dialog box.
- To exclude a field or record field from the tab order, select the Lock the text and activate mouse events if the field is enabled option on the Behavior tab of the Properties dialog box.

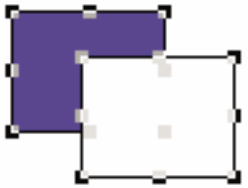
Grouping objects

You can create a group of objects that you work with as a single unit. This allows you to move and format the objects together. For example, you can combine different types of objects, such as a polygon and field, into a group, and then set properties for the group.

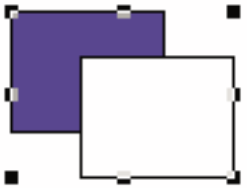
When you create a group, Instructor places it on the lowest layer occupied by an object in the group. For example, if you draw three objects on layers 1, 2, and 3, and then group them, the group is on layer 1. The next object you add will be on layer 2.

➤ **To create a group:**

- SHIFT+click to select each object you want in the group, and then press CTRL+G (or choose Group from the Object menu). After you group the objects, one set of handles surrounds the group.



Two objects not grouped



Two objects grouped

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Editing groups

You can work with a group as a whole, or you can work with the individual objects within the group. For example, you can change properties for individual objects in a group without affecting the properties of the group itself. You can move, size, add objects to, and delete objects from a group without ungrouping it. You can also nest groups within groups.

Before you can edit a group, you must first select it.

➤ **To select a group:**

◆ Click an object within the group.

Any changes you make apply to all of the objects within the group.

➤ **To deselect a group:**

◆ Do one of the following:

— If an entire group is selected, click outside the group's bounding box.

— If an object within a group is selected, double-click outside the bounding box to deselect the entire group.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Editing objects within a group

You can edit individual objects within a group. Simply double-click the object to select it and then make your changes. You can also edit the text in a field that is part of a group. Double-click to select the field, double-click again to display the insertion point, and then make your changes to the text.

➤ **To select an object within a group:**

- Double-click the object you want to select.

  A bounding box appears around the group, and handles appear around the object you selected. Any changes you make apply only to the selected object.

Click another object to change the selection.

Moving grouped objects

You can move an entire group of objects or individual objects within a group by selecting the group or object and dragging it to a new location. After you move an object in a group, Instructor adjusts the group's bounding box to show the new perimeter of the group.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Adding and deleting objects from a group

You can add objects to a group by pasting them from the Clipboard or by dragging them from the Catalog to the group. When you drag an object from the Catalog to a group, Instructor opens the Select Target dialog box, where you can choose an option to add the object to the group.

➤ **To add an object to a group from the Clipboard:**

1. Select the object you want to cut or copy.
2. From the Edit menu, choose Cut or Copy to place the object on the Clipboard.
3. To identify the group that will contain the pasted object, double-click an object in that group.
4. From the Edit menu, choose Paste.

   The object is added to the group. The new object occupies the uppermost layer of the group and is pasted in the same location that it occupied before being cut or copied.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

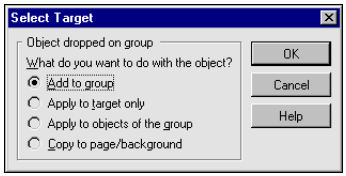
Glossary

Index

➤ To add a Catalog object to a group:

1 Drag an object from the Catalog, and then drop it onto the group.

The Select Target dialog box appears.



2 Choose Add to group, and then click OK.

The group's bounding box expands to encompass the new object. That object occupies the uppermost layer of the group.

Note When dropping objects onto a group, be certain that the mouse pointer is paused over an object in the group. If you attempt to drop onto a space between objects in the group, Instructor will place the object on the page rather than in the group.

After a group is formed, you can delete an object.

➤ To delete an object from a group:

1 Double-click the object to select it.

2 Press DELETE.

The bounding box resizes to reflect the newly defined group.

Note You cannot have fewer than two objects in a group. If you try to delete one of the last two remaining objects, Instructor ignores the Delete command.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Setting group properties

Each group has its own ID number, name, layer, script or action sequence, and drag-and-drop properties. In addition, each member of the group also keeps its own ID number. You can change properties for the group or for individual members of the group. Group properties are set in the same manner as other object properties. For details, see Chapter 7, “Setting object properties.”

Note Selecting the Visible option on the Draw tab of the Properties for Group dialog box does not affect the Visible option of individual objects in the group.

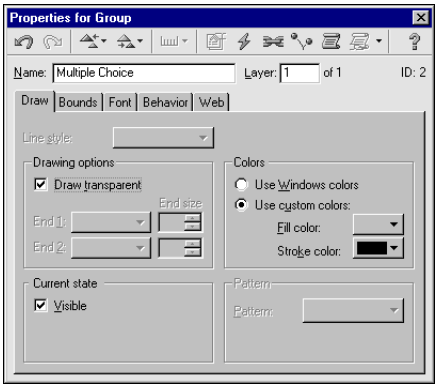


Figure 9
Properties for Group dialog box

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Ungrouping objects

Groups can easily be ungrouped. Ungrouping does not affect individual object ID numbers or names. However, Instructor discards the group's ID number and name.



Important Ungrouping a group will result in the loss of behavior assigned to the group by you or Instructor. Instructor displays a warning if you attempt to ungroup a group with extended behavior.

➤ **To ungroup a group:**

- Select the group you want to ungroup, and then press CTRL+G (or choose Ungroup from the Object menu).

  Instructor replaces the single set of group handles with handles for each object, showing that the objects are now ungrouped.

To cancel the selection of an individual object within a group, hold down the SHIFT key and click the object. The other objects remain selected. To cancel the selection of all the ungrouped objects, click the page away from any selected object.

Instructor places newly ungrouped objects on the layers in front of the old group's layer. For example, if a group contains three objects on layer 1, when you ungroup, the three objects are placed on layers 1, 2, and 3.

Nesting groups

You can make a subgroup of objects within an existing group. The inner (or child) group acts like a single object within the outer (or parent) group. When you ungroup, the child group remains intact. You might want to make child groups when you are creating a complex drawing; when part of the drawing is finished, group it so that you will not accidentally change or move it.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 7

Setting object properties

ToolBook II Instructor provides a rich variety of objects that you can use to create interactive applications quickly and easily. You can set properties for these objects that determine appearance, behavior, and interactive capabilities.

You can also affect the behavior and interactive capabilities of objects using the Actions Editor visual programming tool or the OpenScript programming language. For more information about the Actions Editor, see Chapter 15, "Using the Actions Editor." For more information about OpenScript, refer to the *Programming in OpenScript* online book.

This chapter describes how to set object properties using the Properties dialog box, the Extended Properties dialog box, the right-click menu, and the Property Browser.

IN THIS CHAPTER

Understanding properties and extended properties **180**

Using the Properties dialog box **181**

Using the right-click menu **185**

Using the Extended Properties dialog box **186**

Using the Property Browser **188**



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Understanding properties and extended properties

All ToolBook II Instructor objects (pages, backgrounds, fields, draw objects, buttons, hotwords, and so forth) have basic properties that define their appearance and behavior. While Instructor sets a few properties, such as an object's ID number, most are set by you. The properties you set for a button, for example, might include its fill color, button graphic, and caption.

Many of the objects in the Catalog have additional properties that set more sophisticated capabilities. These properties are called extended properties. A media player, for example, has extended properties that control the player's interactive functionality, such as which media clip will play and how Instructor will play the clip at Reader level.

Properties dialog box

Extended Properties dialog box

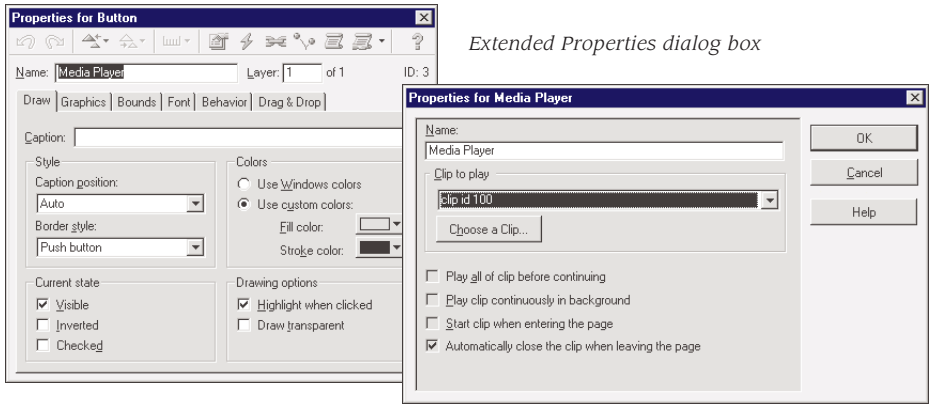


Figure 1
The Properties and Extended Properties dialog boxes for a media player

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the Properties dialog box

The Properties dialog box is a tabbed dialog box that allows you to view and edit an object's properties. It features a toolbar that provides easy access to other dialog boxes, where you can modify the object even further.

The Properties dialog box is quite versatile, because the available settings change dynamically according to the currently selected object. You can leave the Properties dialog box open, select various objects in your book using your mouse, and the settings in the Properties dialog box will change to reflect the available settings for the selected object.

For example, if you select a text field, you will see the Properties for Text Field options in the Properties dialog box. If you select a button, you will see the Properties for Button options in the Properties dialog box. Settings made in the Properties dialog box take place immediately, without closing the dialog box.

➤ **To display the Properties dialog box:**

◆ Do one of the following:

— From the Object menu, choose one of the properties options. (For example, if a button is selected, choose Properties for Button.)



— Click the Properties button on the Instructor toolbar.



— Right-click an object, and then click the Properties button on the right-click toolbar.

— Press SHIFT+F6.

In the Properties dialog box, common properties are organized by tab. The following table describes some of the property pages that are common to many objects.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Examples of property pages common to many objects

Click this tab...	To do this...
Draw	Set the properties that control the way the object is drawn, such as fill color, line style, color palette, or border style
Bounds	Specify the position and size of the object
Font	Set character formatting information for text fields, record fields, and buttons
Paragraph	Set paragraph, alignment, tabs, and spacing information for fields
Behavior	Specify the object's behavior at run time, such as whether a user can type in a field or whether a button is enabled at run time
Drag & Drop	Specify the appearance and behavior of the object when it is dragged or dropped at Reader level

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set an object's properties:**



1 Select the object, and then click the Properties button on the Instructor toolbar.

2 Click a tab to view the properties page containing the category of property you want to set.

3 Choose or enter the properties or settings you want. If you type a value in a box, you must press ENTER, press TAB, or click in another field before the changes will take effect.

4 *Optional*. Click a button on the toolbar to set other available properties of the selected object, such as Extended Properties, Hyperlink, and Path Animation.

More information about the buttons on the Properties dialog box toolbar is available in the following section.

5 *Optional*. Close the Properties dialog box.



Tip After you customize an object by setting its properties, you can copy it to the My Objects category of the Catalog and use it again later. Copying an object to the My Objects category allows you to easily share and reuse custom objects among your Instructor applications. For details, see Chapter 5, "Working with the Catalog."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Toolbar buttons in the Properties dialog box

The Properties dialog box, like the main window, has a toolbar. Buttons on the toolbar become available depending on the object's features and specifications on each tab.

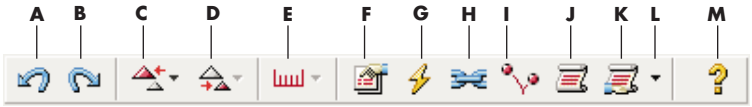


Figure 2
The toolbar in the Properties dialog box

A **Undo** Reverses the last command.

B **Redo** Reapplies the last command.

C **Edit Parent** Changes the displayed properties to the parent of the currently selected object. For example, if you have selected one button in a grouped navigation object, clicking Edit Parent allows you to set properties for the group.

D **Edit Child** Changes the displayed properties to the child of the currently selected object.

E **Set Units** Allows you to choose by what units the object is measured (pixels, page units, inches, or centimeters).

F **Extended Properties** Opens the Extended Properties dialog box for the selected object.

G **Actions** Opens the Actions Editor, where you can program a series of behaviors (called action sequences) for the selected object.

H **Hyperlink** Opens the Hyperlink dialog box, where you can specify a hyperlink for the selected object.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

I **Path Animation** Opens the ToolBook II Animation Editor, where you can create an animation.

J **Edit Script** Opens the script editor, where you can write or edit a script for the selected object.

K **Edit Shared Script** Opens the script editor, where you can edit a shared script.

L **Select Shared Script** Allows you to choose a shared script to edit or to create a new shared script.

M **Help** Opens the online Help topic for the currently displayed tab in the Properties dialog box.



When you have finished setting properties, you can click the Close button in the upper-right corner of the dialog box.

Using the right-click menu

You can set properties for an object by using the right-click menu. The right-click menu contains commands specific to the object you right-clicked; these commands offer a quick way to set properties that you can also set in the Properties dialog box. For example, to name a button, simply right-click the button and choose Name from the right-click menu. The Name dialog box opens, where you can enter a name. The name you specify is set as a property of the button and will appear in the Properties dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

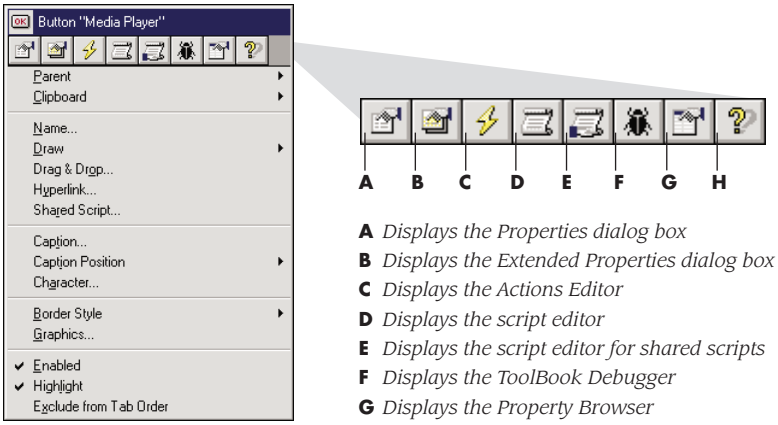
PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



A *Displays the Properties dialog box*
B *Displays the Extended Properties dialog box*
C *Displays the Actions Editor*
D *Displays the script editor*
E *Displays the script editor for shared scripts*
F *Displays the ToolBook Debugger*
G *Displays the Property Browser*
H *Displays context-sensitive online Help*

Figure 3
Sample right-click menu and the right-click menu toolbar

Using the Extended Properties dialog box

You can set the extended properties of Catalog objects in the Extended Properties dialog box. To tell quickly whether an object has extended properties, select the object on the page or background and look at the Extended Properties button on the toolbar. The button is available only when you select an object with extended properties.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set extended properties for an object:**

1 Do one of the following:


— Select the object, and then click the Extended Properties button on the Instructor toolbar.


— Right-click the object, and then click the Extended Properties button on the right-click toolbar.


— In the Properties dialog box for the object, click the Extended Properties button on the toolbar.

2 In the object's Extended Properties dialog box, specify the options you want.

3 Click OK.

Setting extended properties for groups

Many of the objects in the Catalog are actually groups of individual objects. In some cases, the group itself (such as a video player with controls) has extended properties, and in other cases, the individual objects within the group (such as the buttons in a navigation panel) have extended properties.

The easiest way to set extended properties for an entire group is to select the group and then click the Extended Properties button on the Instructor toolbar. To set the extended properties for a single object within a group, double-click to select the object and then click the Extended Properties button on the Instructor toolbar.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

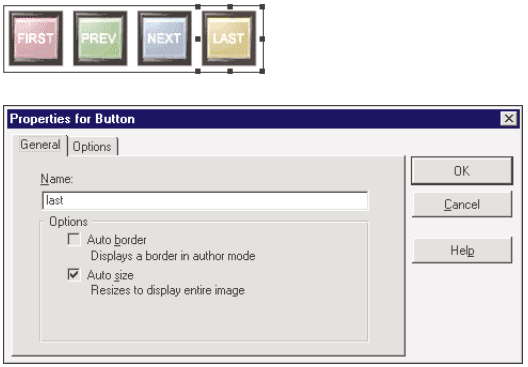


Figure 4
A selected button object and its Extended Properties dialog box

Using the Property Browser

The Property Browser allows you to view and edit a complete list of an object's property settings. In addition, you can view and edit system properties and variables using the Property Browser. System properties define default behavior for the entire Instructor system.

Because properties are displayed in the Property Browser by their OpenScript property name, you should be familiar with the OpenScript language before using this tool. For more information about OpenScript, refer to the *Programming in OpenScript* online book.

Some of the properties that appear in the Property Browser can also be set using either the Properties dialog box or the Extended Properties dialog box. You may prefer to use these dialog boxes instead of the Property Browser to set object properties. However, the Property Browser is the best way to view all of the available properties for an object.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

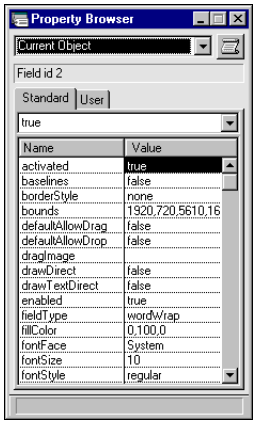
Glossary

Index

➤ **To change a property's value in the Property Browser:**

1 From the View menu, point to Browsers, and then choose Property.

The Property Browser appears.



2 Select an object on your page, or choose the object whose properties you want to edit from the drop-down list at the top of the Property Browser.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

3 Select the property you want to edit from the Name column. The object's current value for that property appears in the Value column and in the drop-down list at the top of the tab.

- If the property uses a specific set of predefined values, click the drop-down arrow, and then choose a different property value.



- If the property does not use a specific set of predefined values, click the Browse button, type a new value in the Edit Property dialog box, and then click OK. Click the check mark button to accept the changes, or click the X button to reject the changes.
- Some properties are for your reference only. You will not be able to edit these properties.



Tip You can also double-click the entry in the Value column to edit the property.

The User tab of the Property Browser displays some properties that are built in to Instructor. You can add a new property for an object by specifying a user property in the Property Browser.

➤ **To add a user property in the Property Browser:**

1 From the View menu, point to Browser, and then choose Property.

The Property Browser appears.

2 Select an object on your page, or choose the object whose properties you want to edit from the drop-down list at the top of the Property Browser.

3 Click the User tab.

4 In the Name column, double-click Add New.

5 In the Add User Property dialog box, type a property name and property value, and then click OK.

Instructor adds the property to the property list.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 8

Working with text

You can add text to your book using field objects, such as text and record fields. Field objects on the pages and backgrounds of your book can be used to convey information and instructions. You can also add inline graphics to field objects to create visual interest within text. This chapter describes how to create, format, and use text in fields and record fields.

IN THIS CHAPTER

About fields **192**

Working with fields **194**

Creating fields **195**

Setting field properties **197**

Adding text to fields **201**

Selecting and editing text **205**

Formatting text in fields **209**

Inserting inline graphics **214**

Inserting special characters **216**

Finding and replacing text **218**

Working with record fields **220**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About fields

Fields are objects that hold text. Before you can add text to your book, you must add a field to contain that text. Instructor offers two basic types of fields: text fields and record fields. A text field can be placed on a page or on a background. If you place a text field on the background, the field and the text it contains appear identically on every page that shares that background. A record field must be placed on a background; unlike a text field placed on the background, a record field can display different text on every page that shares that background.

Consider the following guidelines when adding fields to your book:

- To display different text in the same position and in the same style on every page that shares a particular background, use a record field placed on the background.
- To display identical text in the same position and in the same style on every page that shares a particular background, use a text field placed on the background.
- To display text that appears on one particular page, use a text field placed on the page (the foreground).

The Catalog provides a wide variety of text fields and record fields that you can use to present information on the screen. To add text to your page, you can drag the type of field you want from the Catalog to your page. Then you can replace its placeholder text with your own text.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



Figure 1
Text fields in the Catalog

Text fields and record fields, like other objects, have properties that you can set to define their physical appearance, such as border style, font, and fill color. In addition, some fields in the Catalog have extended properties that you can set. These properties define the extended capabilities of a field, such as automatic sizing, which you set in the object's Extended Properties dialog box. Together, these properties of a text object define its appearance and interactive behavior in your application.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with fields

Like other objects, text and record fields have properties that define their appearance and behavior. After you add a field to a page, you can type or import text into it, and then edit, format, or apply a color to the text. You can also resize and move the field.

Fields can contain graphics and hotwords. For example, you can insert a graphic, such as a bitmap, into a field to create an inline graphic. You can also select a word or phrase and make it a hotword. Hotwords act as buttons: They can trigger an action, such as navigating to a page or playing an animation. For details about creating hotwords, see Chapter 10, "Working with buttons and hotwords."

For the most part, you work with record fields in the same way that you work with text fields. However, take care when cutting, pasting, or deleting record fields. If you cut a record field from the background and paste it on the page, it becomes a regular text field. In addition, removing the record field from the background of one page removes it (and the data it contains) from all pages that share that background. Instructor displays a warning if you attempt to remove a record field.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating fields

There are three ways to create a text or record field. You can drag a field from the Catalog to your page. Alternatively, you can create a field using the Instructor tool palette. Finally, you can use OpenScript to draw a field.

Adding a field from the Catalog

The Catalog provides a wide variety of fields that you can use to present text on the screen. In addition to basic text fields, you can choose special fields that display shadowed text, formatted lists, or a marquee.

➤ **To add a field from the Catalog:**



1 Click the Catalog button on the Instructor toolbar to open the Catalog.

2 Select the Text Field category in the Catalog.

3 Choose the field you want from the object pane, and then drop it onto your page or background.



Tip You can also double-click an object in the Catalog to add it to your page.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating a field using the tool palette

There are three tools on the tool palette that you can use to draw Instructor field objects:



- **Field tool** Use the field tool to draw a field. You can draw a field on the page (foreground) or background.



- **Borderless field tool** Use the borderless field tool to draw a field without a border. You can draw a borderless field on the page or background.



- **Record field tool** Use the record field tool to draw a record field on a background. This tool is disabled when you are working on the page and becomes available when you switch to the background.

➤ **To create a field using the tool palette:**

- On the tool palette, choose one of the field tools. On your page, hold down the mouse button as you drag to draw a field.

Creating a field using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

- Use the *draw* command to draw a field or record field. Fields are measured in page units. For example:

```
draw field from 1000,500 to 2000,3000
draw recordField from 1000,500 to 2000,3000
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Setting field properties

You can select a field and then set its properties in the Properties for Text Field or Properties for Record Field dialog box. Each tab in these dialog boxes contains settings that allow you to specify some aspect of a field's behavior and appearance.

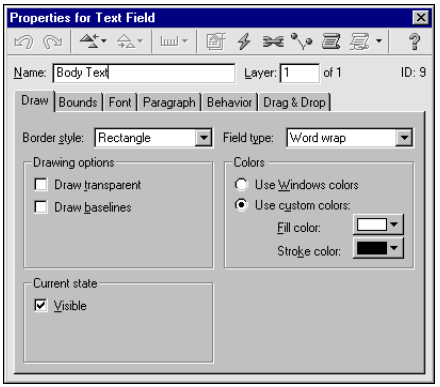


Figure 2
Properties for Text Field dialog box

Fields are identified by the same properties that identify other objects, such as a name and ID number. Additional properties define specific characteristics of fields. The field type property defines the type and function of a field—for example, whether it is a wordwrap text field or a multiple-select list box. A field also has properties that define its behavior at run time, such as whether a field is editable (allows users to enter text) or activated (locks the text and activates mouse events) at Reader level.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set the properties of a field:**



1 Select the field, and then click the Properties button on the Instructor toolbar.

The Properties for Text Field or Properties for Record Field dialog box appears.

2 Click one or more tabs and select or enter the options you want to set, as described in the table below.

Instructor field properties

Click this tab...	To do this...
Draw	Set drawing options, border style, field type, color, and visibility state options
Bounds	Set the field's position and size
Font	Set the font, style, size, effects, and formatting options
Paragraph	Set text alignment, indentation, spacing, and tabs
Behavior	Specify the field's behavior at run time
Drag & Drop	Specify the appearance and behavior of the field when it is dragged or dropped at Reader level

3 *Optional*. Close the Properties dialog box.

You can also use the Actions Editor visual programming tool to create action sequences that set certain field properties while the application is running. For more information about the Actions Editor, see Chapter 15, "Using the Actions Editor," and Chapter 16, "Creating action sequences: Practical examples."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Modifying border style and field type

You can modify the basic appearance and function of a field by changing your selections in the Border style and Field type lists (both appear on the Draw tab of the Properties for Text Field or Properties for Record Field dialog box). For example, you can apply a number of different border styles to a field, as shown in Figure 3.

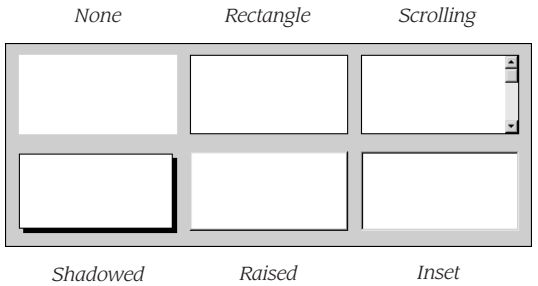


Figure 3
Field border styles

You can change the way text wraps within a field using the options in the Field type list in the Properties dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Field type options

Option	OpenScript value	Result when applied
Wordwrap	*wordWrap*	Text wraps to the next line.
No wordwrap	*noWrap*	Text does not wrap to the next line; text longer than the width of the field truncates at the field's border. Pressing the ENTER key, however, creates a carriage return/linefeed (CRLF)—that is, it starts a new line.
Single line	*singleLineWrap*	Limits text to a single line; text won't wrap, and a CRLF isn't recognized.
Single-select list box	*singleSelect*	Limits user to selecting only one item at a time from the text options in a list box.
Multi-select list box	*multiSelect*	Allows user to select multiple items in a list box.

Modifying the behavior of a field

You can set the properties of a field that define its behavior at Reader level. For example, a field can either be editable or activated. When a field is editable (the field allows typing), a user can enter text in the field at Reader level. When a field is activated (the field locks the text and activates mouse events), a user cannot edit the text. Instead, the field responds to a mouse click the same way a button would: It performs a defined action. For example, if a user clicks a hotword (containing a defined hyperlink) in an activated field, Instructor navigates to a specified page. For more information about hotwords, see Chapter 10, "Working with buttons and hotwords."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The following table describes the options available in the Properties dialog box that define the behavior of a field at Reader level.

Field behavior options

Option	Result when applied
Enable the field	The field can receive the focus or mouse-event messages at Reader level.
Allow users to enter or modify text	The text in a field can be edited by users at Reader level. This option is available when a field is enabled.
Lock the text and activate mouse events	The text in a field cannot be edited by users at Reader level. Mouse events are activated (for example, a hotword's hyperlinks can be run). This option is available when a field is enabled.

Adding text to fields

To add text to the pages of your application, you type or import the text into a text or record field and format it using the options in the Properties dialog box. You can also use the Clipboard to transfer text from another application or between books, and incorporate preformatted text in Rich Text Format. For details, see "Cutting, copying, and pasting text," later in this chapter.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Typing text

To select or enter text, double-click a field to switch Instructor from object-selection mode to text-editing mode. You can also switch to text-editing mode by selecting a field and then choosing Edit Text from the Text menu.

➤ **To type text:**

Using the interface

1 Double-click the field.

2 If the field contains placeholder text, select the text that you want to replace, and then type your text.

3 To end text-editing mode, click outside the field or press the ESC key.

Using OpenScript

◆ To enter text in a field, you set the *text* property of that field:

```
text of field "city" = "Yakima"
```



Tip After the insertion point appears in one editable field, you can place the point in another editable field by clicking only once.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Importing text

You can add text to your book from other programs by importing a file or by copying and pasting a selection of text. When you add text from another program, most character formatting is kept, and paragraph formatting used in the first paragraph is applied to subsequent paragraphs in the selection.

Instructor allows you to import text in Rich Text Format (RTF) or in ASCII (American Standard Code for Information Interchange) format. Most text-editing applications, such as Microsoft Word, allow you to save your text in either of these formats.

➤ **To import text:**

1. Double-click the field into which you want to import text.
2. From the Insert menu, choose Text.

   The Import Text dialog box appears.
3. In the List files of type list, select an RTF file (with an .rtf file extension) or an ASCII text file (with a .txt file extension), and then click OK.

   If your field already contains text, Instructor notifies you that the current contents of your field will be replaced.
4. Click OK to continue, or click Cancel to preserve your existing content.



Note When you import text, Instructor replaces the entire contents of the field. To avoid losing existing text, you can always import the text into an empty field and copy and paste the text to the location you want.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Copying and pasting text from an RTF document

When you copy and paste text into Instructor from other programs that support RTF, the text will retain most of its formatting. Paragraph formatting applied to the first paragraph is applied to subsequent paragraphs in the selection.



Note When using RTF text, you may see paragraph formatting in the source application that does not appear in Instructor. Although some formatting may be lost, no text data is lost.

➤ **To paste RTF text into Instructor:**

1. In the source application, select the text you want, and then copy it to the Clipboard.
2. In Instructor, double-click the field into which you want to paste the text.
3. From the Edit menu, choose Paste.

   Instructor pastes the text into the field or record field.

You can paste RTF text into other programs from Instructor via the Clipboard using the Copy and Paste commands on the Edit menu. When you copy text, Instructor places RTF text on the Clipboard. If the destination program supports RTF, it uses this format when you paste the text. The text retains the formatting that the destination program can recognize; for example, Instructor exports color information for text, but some programs can't use color.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Selecting and editing text

You can select and edit the text in a field. For example, you can use the Clipboard to cut, copy, or paste text in a field. The following sections describe the basic techniques you use when working with text.

Navigating in text

You can double-click a field to create an I-beam called the insertion point, which you use to indicate where you want text to be inserted when you type. You can click with the mouse to move the insertion point within any text in a field or to another field on the same page.

The following table describes the keyboard shortcuts you can use to navigate in a text field.

Navigating in text

To move the insertion point...	Press this key or key combination...
One character left or right	RIGHT ARROW or LEFT ARROW
One text line up or down	UP ARROW or DOWN ARROW
To the start of the text line	HOME
To the end of the text line	END
To the beginning of the text	CTRL+HOME
To the end of the text	CTRL+END
Within a large text block	PAGE UP or PAGE DOWN

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Selecting text

Before you can change the appearance of text, you must select it. Selecting highlights just the text with which you want to work.

Selecting text

To...	Do this...
Show the insertion point	Double-click the field.
Select a character	Click and drag over the character.
Select a word	Double-click the word.
Extend a selection	Hold down the SHIFT key and press a navigation key to move the insertion point.
Select all the text	Press CTRL+A.

Cutting, copying, and pasting text

You can cut or copy an entire field or any selected text in a field. You can cut, copy, and paste text from one field to another.



Tip You can also cut, copy, and paste fields or selected text using the right-click menu or using the standard Windows shortcut keys CTRL+X, CTRL+C, and CTRL+V.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To cut or copy text:

Using the interface

1 Select the text you want to cut or copy.

2 From the Edit menu, choose Cut or Copy.

Instructor places the selected text on the Clipboard.

Using OpenScript

- To copy the text, fonts, and styles of one field to another, set the *richText* property of the field you want to receive the formats to the *richText* of the field containing the desired formats:

```
richText of field "rats" = richText of field "mice"
```

- To copy the text of one field to another without copying the formatting, set the *text* property of one field to the *text* property of the other:

```
text of field "rats" = text of field "mice"
```

- To copy the RTF codes separately from the text, set the *text* property of one field to the *richText* property of another:

```
text of field "rats" = richText of field "mice"
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To paste text:**

Using the interface

1 Copy the text you want to the Clipboard.

2 Double-click the field into which you want to paste text.

An insertion point appears in the upper-left corner.

3 From the Edit menu, choose Paste.

4 To end text-editing mode, click outside the field or press the ESC key.

Using OpenScript

◆ You can use the *send paste* statement to paste the contents of the Clipboard into a new field.

Deleting text

You can remove text from a field. If you delete text by mistake, immediately choose Undo from the Edit menu to restore it; otherwise, your text will be lost.

➤ **To delete text:**

1 Select the text you want to delete.

2 Press DELETE, press BACKSPACE, or choose Delete Selected Text from the Edit menu.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Formatting text in fields

You can apply character formatting (such as font, style, color, and size) and paragraph formatting (such as alignment, line spacing, indents, and tab settings) to the text in a field using the Font and Paragraph tabs on the Properties dialog box.

➤ **To use the Properties dialog box to change text and paragraph formatting for the entire field:**

Using the interface



1 Select the field, and then click the Properties button on the Instructor toolbar.

The Properties dialog box appears.

2 Do one or both of the following:

- To change the font, style, size, or other text effects, click the Font tab, and then select the options you want.
- To change the alignment, line spacing, indents, or tab settings, click the Paragraph tab, and then select the options you want.

3 *Optional.* Close the Properties dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- To change character style for the entire field using OpenScript, specify the field name:

```
fontStyle of text of field "status" = bold
```

- To change the character style used in all fields on a page, select the fields and set the font:

```
select all field
fontFace of selection = "Arial"
fontStyle of selection = "bold"
```

- To change the default text style, set the *sysFontFace, sysFontStyle,* and *sysFontSize* properties:

```
sysFontFace = "Arial"
sysFontStyle = "bold"
```

- You can set the *textAlignment*, *spacing*, *indents*, *tabType*, and *tabSpacing* field properties to modify paragraph formats. For example, to set line spacing in a field:

```
spacing of field "employees" = 2
```

You can format specific characters by selecting them and applying character styles to them. You can also change the character style of an entire field, or set a default character style so that all of the text you enter has a consistent format. Your default character style remains in effect for the current session.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To format specific characters in a field:**

Using the interface

1 Select the text you want to format.



2 Click the Character button on the Instructor toolbar.

The Character dialog box appears.

3 Choose the font, style, size, and effects you want.

4 Click OK.

Using OpenScript

◆ To change character style of specific characters in a field using OpenScript, select the text:

```
select textline 1 of text of field "status"
fontSize of selectedText = 12
```

Applying color to text

Instructor uses the current stroke and fill colors when you draw a field or enter text in it. The stroke color determines the color of the text and the border, and the fill color defines the background color. You change the color of the text and background the same way you change the color of any object—in the Properties dialog box. For details about coloring objects, see Chapter 6, "Working with objects." You can also apply colors to specific characters.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To format selected text with color:**

Using the interface

1 Select the text you want to change.



2 Click the Color Tray button on the Instructor toolbar.

3 Click the Stroke Color button.

4 Choose the colors you want for the selected text.

Using OpenScript

- The *strokeColor* property defines the default color of text in a field or record field. The *fillColor* property defines the color of the area within the field. You can specify a color by its OpenScript keyword, such as *yellow* or *green,* or you can use RGB (red, green, blue) or HLS (hue, lightness, saturation) values. For example:

```
strokeColor of selectedText = blue
fillColor of field "address" = yellow
rgbStroke of selection = 0,255,0          --Sets text color to green
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Formatting paragraph styles

By changing a field's paragraph settings, you can change the appearance of the field and the text it contains. Paragraph style always applies to an entire field (unlike character style, which can be used to format specific characters). You can change:

- text alignment.
- spacing between lines of text.
- tab settings for left-aligned or decimal-aligned text.
- left, right, and first-line indents (starting at the edge of the field).



Note Paragraph style applies to every instance of a record field; if you center text in a record field on one page, the text in that record field is centered on every page.

➤ **To format paragraph style:**

Using the interface



1 Select the field, and then click the Properties button on the Instructor toolbar.

   The Properties dialog box appears.

2 Click the Paragraph tab, and then select the settings you want for alignment, line spacing, tabs, and indents.

3 *Optional*. Close the Properties dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

- You can set the *textAlignment*, *spacing*, *indents*, *tabType*, and *tabSpacing* field properties to modify paragraph formats. For example, to set line spacing in a field:

```
spacing of field "employees" = 2
```

Inserting inline graphics

You can insert an inline graphic anywhere in a field. Inline graphics are graphics that are set within text and can include bitmaps, icons, or cursors. When you add an inline graphic, the graphic is automatically added to the book's resources. For more information about book resources in Instructor, see Chapter 17, "Using resources."



Tip You can use inline graphics as hotwords. To create a hotword from an inline graphic, create the inline graphic, and then select it. Then, from the Text menu, choose Create Hotword. After you create the hotword, you can right-click it, choose Hyperlink, and set its hyperlink properties.

You can cut, copy, paste, or delete inline graphics just like any character. Text formatted in RTF that contains bitmaps can be pasted into fields. The bitmaps become inline graphics. In addition, you can drag bitmaps and other graphic resources from the Media category of the Catalog to a field to quickly create an inline graphic.

Inline graphics wrap to the next line as you add text, and the bottom of each graphic is aligned along the text's baseline. To add space above or below an inline graphic, use the Superscript or Subscript command on the Text menu.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To insert an inline graphic:**

Using the interface

1 Double-click the field to display the insertion point, and then move the insertion point to where you want to insert the graphic.

2 Do one of the following:

- Drag a graphic object from the Media category of the Catalog to the field.
- From the Text menu, choose Insert Graphic, choose or import the graphic file you want, and then click OK.

Instructor inserts the graphic into the field.

Using OpenScript

- Use the *insert graphic* command to insert the graphic into a field:

```
insert graphic bitmap "Fog" into char 1 of textline 3 of \
   text of field "Horn"
insert graphic icon "speaker" into last char of text of \
   hotword "sound"
insert icon ID 100 after text of field ID 1
insert graphic bitmap "Fog" before text of focus
insert graphic cursor "wait" into char 2 of selectedText
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Inserting special characters

You can also insert ANSI (American National Standards Institute) characters to create curled single and double quotation marks (‘ or “), em dashes (—), copyright (©) or trademark (™) symbols, and other helpful characters. To type ANSI characters, hold down the ALT key, and then—using the numeric keypad—type *0* and the ANSI number that corresponds to the character you want. For example, to type a copyright symbol (©), hold down the ALT key and type *0169*.



Note Some font styles do not support the ANSI character set. You may need to change fonts in order to use some ANSI characters.

The table on the following page displays the ANSI character set.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The ANSI extended character set

0	†	32	†	64	@	96	`	128	§	160		192	À	224	à
1	†	33	!	65	A	97	a	129	§	161	¡	193	Á	225	á
2	†	34	"	66	B	98	b	130	‚	162	¢	194	Â	226	â
3	†	35	#	67	C	99	c	131	*f*	163	£	195	Ã	227	ã
4	†	36	$	68	D	100	d	132	„	164	¤	196	Ä	228	ä
5	†	37	%	69	E	101	e	133		165	¥	197	Å	229	å
6	†	38	&	70	F	102	f	134	†	166	¦	198	Æ	230	æ
7	†	39	'	71	G	103	g	135	‡	167	§	199	Ç	231	ç
8	‡	40	(	72	H	104	h	136	ˆ	168	¨	200	È	232	è
9	‡	41	)	73	I	105	i	137	‰	169	©	201	É	233	é
10	‡	42	*	74	J	106	j	138	Š	170	ª	202	Ê	234	ê
11	†	43	+	75	K	107	k	139	‹	171	«	203	Ë	235	ë
12	†	44	,	76	L	108	l	140	Œ	172	¬	204	Ì	236	ì
13	‡	45	-	77	M	109	m	141	§	173	-	205	Í	237	í
14	†	46	.	78	N	110	n	142	§	174	®	206	Î	238	î
15	†	47	/	79	O	111	o	143	§	175	¯	207	Ï	239	ï
16	†	48	0	80	P	112	p	144	§	176	°	208	Ð	240	ð
17	†	49	1	81	Q	113	q	145	‘	177	±	209	Ñ	241	ñ
18	†	50	2	82	R	114	r	146	’	178	²	210	Ò	242	ò
19	†	51	3	83	S	115	s	147	“	179	³	211	Ó	243	ó
20	†	52	4	84	T	116	t	148	”	180	´	212	Ô	244	ô
21	†	53	5	85	U	117	u	149	•	181	µ	213	Õ	245	õ
22	†	54	6	86	V	118	v	150	–	182	¶	214	Ö	246	ö
23	†	55	7	87	W	119	w	151	—	183	·	215	×	247	÷
24	†	56	8	88	X	120	x	152	˜	184	¸	216	Ø	248	ø
25	†	57	9	89	Y	121	y	153	™	185	¹	217	Ù	249	ù
26	†	58	:	90	Z	122	z	154	š	186	º	218	Ú	250	ú
27	†	59	;	91	[	123	{	155	›	187	»	219	Û	251	û
28	†	60	<	92	\	124	|	156	œ	188	¼	220	Ü	252	ü
29	†	61	=	93	]	125	}	157	§	189	½	221	Ý	253	ý
30	†	62	>	94	^	126	~	158	§	190	¾	222	Þ	254	þ
31	†	63	?	95	_	127	§	159	Ÿ	191	¿	223	ß	255	ÿ

† No conversion for this character.
‡ Values 8, 9, 10, and 13 convert to backspace, tab, linefeed, and carriage return, respectively.
§ Values 127–129, 141–144, and 157–158 vary depending on the font manufacturer.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Finding and replacing text

You can use the Find command on the Edit menu to search for text, and the Replace command if you also want to specify replacement text.

Instructor searches the entire book starting on the current page and prompts you to start over when it reaches the end of the book. If you want to limit the search, you can select one of the options in the Find dialog box. You can search for text in:

- all fields and list boxes shown on the current page.
- record fields on all pages that share the same background.

You can also customize your search using the Match whole word only option. For example, if you specify “the” in the Find what box, Instructor ignores “these.” Or you can use the Match case option to find only text that matches the capitalization you entered in the Find what box. For example, if you enter “Hello World,” Instructor ignores “Hello world.”

Fields that you have hidden cannot be searched using the Find or Replace commands. However, Instructor does search fields that are obscured by another object (including the Find dialog box) or are outside the bounds of the window. Instructor scrolls the text in a field to display found text, and the Find button changes to Find Next.



To be sure you will see all found text, choose Size to Page from the Page menu, or click the Maximize button before using the Find or Replace command.

If your application uses the Find and Replace commands at Reader level, users of your application will be able to search fields and record fields. Instructor can find text in activated fields, but at Reader level this text cannot be edited.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

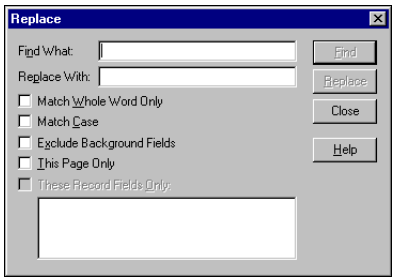
PART VI
Distributing and printing your application

Glossary

Index

➤ To find or replace text:

1 From the Edit menu, choose Find or Replace.



2 Enter the text for which you want to search in the Find what box. If you are replacing text, enter the new text in the Replace with box.

3 Choose search options, if appropriate.

You can choose to match a whole word and to match uppercase and lowercase text.

4 Click Find to start the search.

Instructor finds and highlights the first occurrence of the text. If the text is on a different page, Instructor goes to the page and then highlights the text.

5 To replace the text, click Replace.

6 Click Find Next to find the next occurrence of the text.

When Instructor reaches the end of the book, you are prompted to continue searching from the beginning of the book. Click Yes or No.

When you finish searching, close the Find or Replace dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with record fields

You can add record fields to your book using the record field tool on the Instructor tool palette. Because you can place a record field only on the background of a page, you will notice that this tool is disabled when you are working on the foreground.

Like any object placed on a background, a record field appears in the same style and in the same position on any page that shares that background. However, unlike a regular text field placed on the background, a record field allows you to enter and display different text from page to page. Instructor can also sort pages and print reports using record fields.

Using record fields to display information

Record fields are ideal for consistently presenting information that varies from page to page, such as address and phone lists or client contact records. After you've created the record field on the background, you can enter text into it on any page that shares that background by switching to the foreground (press F4), double-clicking the record field, and then typing your text. You can type different text into the record field on each page. For example, if you have a record field on a background, and that background is shared by five pages, you can type different text into each of the record fields; Instructor will then display different text on all five pages.



Tip To prevent users from typing text into a record field in which you're displaying text or inline graphics, in the Properties for Record Field dialog box, select the Lock the text and activate mouse events if the field is enabled check box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

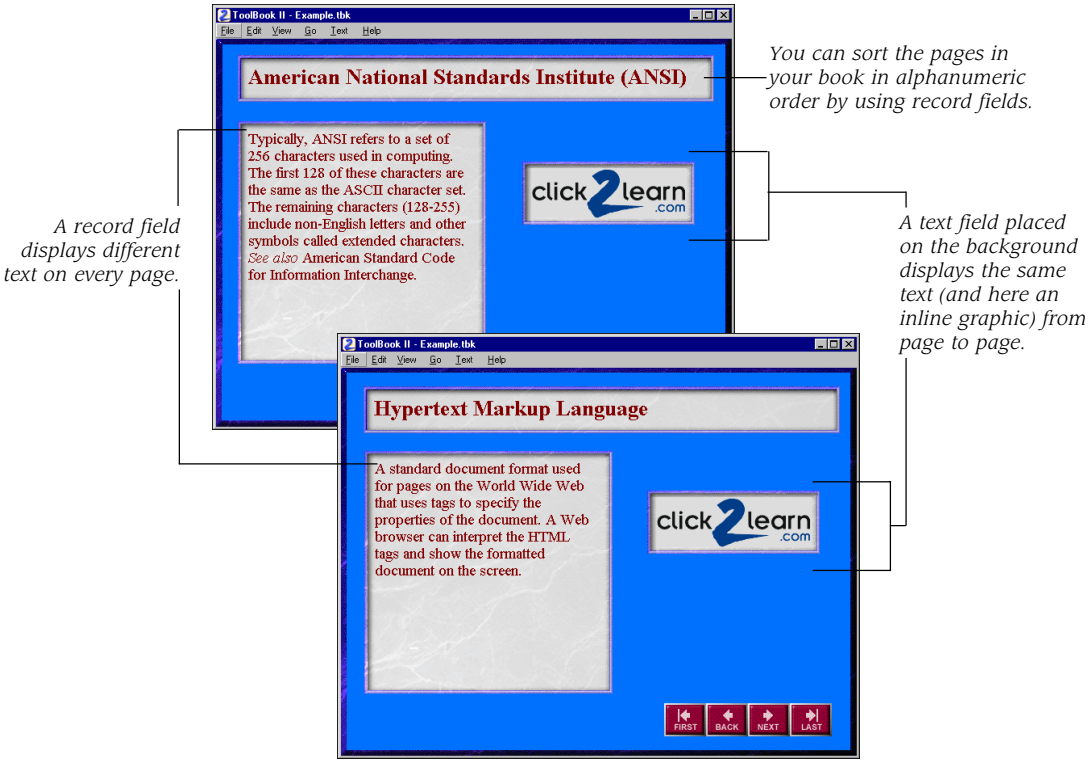


Figure 4
Use record fields to create an online glossary

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using record fields for Reader-level data entry

You can use record fields to create a form into which your users enter data at Reader level. For example, you might set up a product catalog to index all the software products used at your company. As shown in Figure 5, each data-entry field is actually a record field.

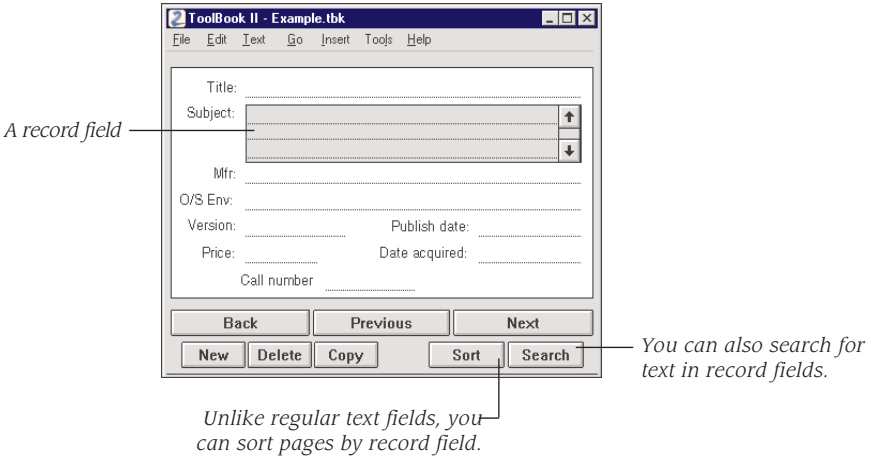


Figure 5
Sample data-entry application

Record fields are a good choice for creating data-entry applications, because Instructor can print reports or sort pages using record fields.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Importing pages with record fields

You can import pages that contain record fields from another book into the current book using the Paste command on the Edit menu or using the OpenScript *import* command. When you import pages, Instructor inserts the pages into the destination book after the current or specified page.



Note If you paste record fields into your book using the Paste command, be sure that you are on the background when you paste; otherwise, Instructor will create text fields from them.

When you import pages and choose not to include the source book's backgrounds, Instructor adds the record field text to the destination book's record field text if the source and destination books' record field names or ID numbers match. If the source and destination books' record field names or ID numbers do not match, the source record field text is added to the destination record field using layer order. The source book's first record field is mapped to the destination book's first record field. If the source book contains more record fields than the destination book, the extra record field text is not mapped to a destination record field.

Sorting pages by record field

Sorting is the process of putting pages in order according to the contents of one or more record fields. Generally, you sort pages to put them in the order in which you expect them to be used. For example, you might sort a library catalog alphabetically by author or title. Use the Sort Pages command on the Tools menu to sort pages using the text of any record field according to the sort type you specify.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Because sorting is based on record fields, only the set of pages that share the same background can be sorted; however, within that set you can specify a range of pages to sort. If your application is designed so that a user can add entries, you can provide sorting at Reader level using the Sort dialog box, or you can design your own sorting option using the OpenScript *sort* command. The *sort* command is much more flexible and can sort across books using any page-based comparison.

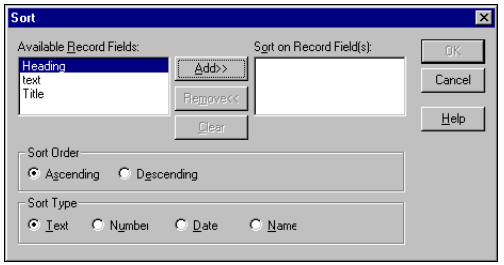
For details about using the *sort* command, refer to its entry in the OpenScript reference in online Help.

➤ To sort pages by record field:

Using the interface

1 From the Tools menu, choose Sort Pages, or press SHIFT+F5.

The Sort dialog box appears.



2 Select the record field by which you want to sort in the Available record fields box, and then click Add.

If you want to add more than one record field to the Sort on record field(s) list, you must select each individually and click Add for each one. Instructor sorts pages using the record fields in the order that you have selected them.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

3 Under Sort order, choose Ascending or Descending.

4 Under Sort type, choose one of the following:

- **Text** Sorts pages in alphanumeric or reverse alphanumeric order using the text entered in the record field.
- **Number** Sorts pages in numeric or reverse numeric order using the numbers entered in the record field.
- **Date** Sorts pages using the date entered in the record field. For example, if you choose Ascending, Instructor puts a page with the date 08/18/68 before a page with the date 02/07/93. Instructor uses the date format set by the *sysDateFormat* property.
- **Name** Sorts pages using the name entered in the record field. Instructor uses the last word in this field for sorting. For example, if you choose Ascending, Instructor puts a page with the name "Zoe Z. Benton" before a page with the name "Angela Victor."

If Instructor encounters data that does not match the type you have specified (for example, text in a field that you're sorting as a date), it places that page at the end.

5 Click OK to sort the pages.

Using OpenScript

- To sort pages, use the *sort* command with any expression that can be evaluated in the context of a page, such as page name or page ID number. Alternatively, specify a range of pages with an expression. For example:

```
sort pages 1 to pageCount of this book by ascending text \
   name of this page
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

When you sort pages by a record field treated as a name, Instructor treats the first word in the first paragraph as a first name, the second word as a middle name, and the third word as a last name. Additional words are ignored. Instructor sorts by last name, then by first name, and finally by middle name or initial. A one-word paragraph is considered a last name; two words are considered to be first and last name, in that order. Two words concatenated with the tilde (~) character are evaluated as one word. For example, “Hubert Van~Beek” is sorted by “Van Beek” but “Hubert Van Beek,” is sorted by “Beek.” The tilde appears only onscreen and is not printed. Any amount of additional text can follow the first paragraph.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 9

Using graphics

This chapter describes how you can use graphics in ToolBook II Instructor to enhance the visual appearance of your application. With Instructor, you can paste or import graphics into your application, drag graphics from the Catalog to your application, and assign resources to the objects in your book. This chapter describes how to add graphics and modify them in your application.

IN THIS CHAPTER

About graphics **228**

Using graphics on the Internet **229**

Adding graphics to your book **229**

Modifying graphics **233**

Creating and applying a color palette **238**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About graphics

You can paste or import graphics created in other programs into your ToolBook II Instructor application. When you paste or import a graphic, Instructor creates one of the following types of graphic objects:

- **Picture object** Instructor stores picture objects in the Windows Metafile Format (WMF), a file format for vector graphics. A vector graphic can be an image comprising a collection of lines or a pixel-based image that is resizable.
- **Paint object** Instructor stores paint objects in the bitmap (BMP) file format. Bitmaps consist of small dots that form an image that is not resizable.

Picture objects contain a richer set of graphical instructions than paint objects. When deciding whether to add a graphic as a picture object or paint object, consider that Instructor usually displays paint objects faster. However, paint objects require more disk space than picture objects.

To identify the graphic type at a glance, right-click the graphic. The right-click menu identifies the type of object.

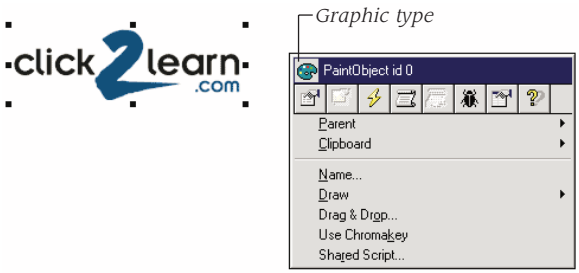


Figure 1
Right-click menu for an imported graphic

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using graphics on the Internet

When you publish your application to the Web, Instructor exports all paint and picture objects to Internet-compatible file formats. Picture objects and 8-bit paint objects are converted to GIF (Graphics Interchange Format) files. Paint objects that contain more than 256 colors will be converted to JPEG (Joint Photographic Experts Group) files. If your book contains bitmaps that were added as resources, all 8-bit bitmap images are converted to GIF files, and 16-bit and 24-bit bitmap images are converted to JPEG files.



Note Images that have a chromakey color set (to make one color invisible) will be exported as GIF files. For best results during conversion of a chromakey image, import the image into ToolBook as a 256-color image with an optimized palette.

Adding graphics to your book

You can incorporate graphics into your Instructor application in several ways. You can:

- use the Clipboard to copy and paste a graphic.
- use the Graphic command on the Insert menu to import a graphic.
- drag a graphic from the Catalog to a page, background, or another object.
- use the Properties dialog box to assign a graphic to a book or button as a resource.

Although you can copy, paste, and insert graphics into your book, there is an advantage to adding graphics as resources. Resources are files that can be used multiple times in your Instructor book. The advantage of using graphics as resources is that your file size does not increase each time you use the same graphic. For more information about using resources, see Chapter 17, “Using resources.”

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Pasting graphics from the Clipboard

You can use the Clipboard to copy a graphic from another application and paste it into Instructor, as well as to copy and paste a graphic from another Instructor book into the current one. In addition, you can paste graphics into text fields and record fields, creating inline graphics. For details about inline graphics, see Chapter 8, "Working with text."

➤ **To copy and paste a picture object or paint object:**

1. Run the application that contains the graphic you want to paste.
2. Select the graphic, and then copy it to the Clipboard.
3. In Instructor, go to the page or background on which you want to paste the graphic.
4. Do one of the following:
   - From the Edit menu, choose Paste. Instructor adds the graphic in the richest file format that the copied graphic will yield.
   - From the Edit menu, choose Paste Special. In the As list, select a file format, and then click OK. Instructor adds the graphic in the file format you selected.

   Instructor places the picture object or paint object in the upper-left corner of the current page or background. The image may be framed with a border. If you don't want a border, go to the Draw tab of the object's Properties dialog box and select None from the Line style list.



Note Be sure to edit or resize a paint object before you paste or import it, because you cannot edit the image in Instructor.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Importing graphics

When you import a graphic using the Graphic command on the Insert menu, Instructor uses an import filter to convert the graphic file to a picture object or paint object.

To import graphics, you must have the appropriate graphic filters installed on your system. An import filter is a program Instructor runs to convert a graphic to a format it can use. The Files of type list in the Import Graphic dialog box shows the filters available to you.

➤ **To import a picture object or paint object:**

Using the interface

1 From the Insert menu, choose Graphic.

Instructor displays the Import Graphic dialog box.

2 Navigate to and select a file from the list.

3 Click Insert.

Instructor imports the graphic object and frames it with a border. If you don't want a border, go to the Draw tab of the object's Properties dialog box and select None from the Line style list.

Using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

◆ To import a graphic, use the *importGraphic* command. Specify the file name and extension, along with the complete path, if necessary. For example:

```
importGraphic "c:\windows\city.bmp"
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using graphics from the Catalog

The Media category of the Catalog contains a wide variety of bitmap graphics, backdrops, icons, and other images that you can use in your application. You simply drag the bitmap or graphic you want from the Catalog to your target object. For example, you can drop graphic objects on:

- pages or backgrounds to create bitmap graphics.
- backgrounds to create tiled backdrops.
- fields to create inline graphics.
- buttons, graphic placeholders, and question objects to change their assigned graphic.

When you use a graphic from the Catalog, Instructor creates a graphic resource, which you can reuse. For more information about using book resources, see Chapter 17, “Using resources.” For more information about the Catalog, see Chapter 5, “Working with the Catalog.”

Assigning a graphic resource to an object

A resource is a file that you can use multiple times in your Instructor application. For example, to make your interface more graphical and consistent with other Windows applications, you can apply bitmaps to various button states (normal, checked, and so forth) using the Properties for Button dialog box. These bitmaps can be stored as resources in a common library, making them available to all other objects in the book that can use that type of resource.

Most of the graphics that you use in Instructor will be stored as resources. For more information about managing graphics as resources, see Chapter 17, “Using resources.”



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Storing graphic files in Instructor

All graphic objects are stored in a ToolBook II proprietary format with persistent properties (such as size and border style). Imported graphic objects have additional information attached that tells Windows what kind of graphic the object is and how to display it. A vector graphic is converted into an Instructor version of the Windows Metafile Format when it is imported into Instructor. This conversion takes place whether the original file was a metafile (with a .cgm or .wmf extension) or in another format, such as TIFF (Tagged Image File Format) or EPS (Encapsulated PostScript).

Modifying graphics

Like other objects, graphics have properties that you can set to modify their appearance or behavior. You can modify an imported graphic in several ways, depending on the type of object. For example, you can:

- crop paint objects by adjusting the bounding box.
- change the line style and color of the border of an imported graphic object.
- convert picture objects to paint objects.
- create transparent areas in paint objects.

The following sections describe the techniques you use to modify objects.

Cropping paint objects

In Instructor, you can adjust the handles around a paint object to show only a portion of the original image.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

If you move the handles of a paint object, Instructor crops the right side of the image to fit within the new bounding box size. If you try to make a paint object larger than its original size, Instructor adds a transparent area around the image. If you move the handles of a picture object, the image is resized.

Original paint object *Cropped paint object*



Figure 2
Cropped paint object

➤ **To crop a paint object:**

Using the interface

1 Select the paint object to display its handles.

2 Drag a handle to display only the desired portion of the image within the bounding box.

Using OpenScript

◆ Adjust the size of a paint object by setting its *bounds* property:

```
bounds of paintobject "flower" = 100, 100, 2000, 2000
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Changing the line style and border color of an imported graphic

After you have imported a graphic, you can apply a line style and a stroke color to its border in the Properties dialog box. Applying a fill color to a bitmap or picture object has no effect; all colors except the border color must be applied or created before you import the graphic into Instructor.

➤ **To change the line style and border color of an imported graphic:**

Using the interface



1 Select the graphic, and then click the Properties button on the Instructor toolbar.

2 In the Properties dialog box, click the Draw tab.

3 In the Line style list, select the line style you want.

4 Click Stroke color.

A color palette opens.

5 From the color palette, choose the color you want.

Instructor applies the color to the graphic object's border.

6 *Optional.* Close the Properties dialog box.

Using OpenScript

- An imported graphic object is framed with a border set to the current value of *sysLineStyle*. If you do not want a border, set the object's *lineStyle* to *none*. To change the border's color, specify values for the object's *strokeColor*.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Converting picture objects to paint objects

In some cases, you might want to convert a picture object to a paint object. For example, you might want to improve the performance of an application that contains many picture objects, or you might want to crop an image originally imported as a picture object.

Important This conversion cannot be reversed. Make a backup copy of the picture object before you convert it to a paint object.

➤ **To convert a picture object to a paint object:**

Using the interface



1 Select the picture object, and then click the Properties button on the Instructor toolbar.

2 In the Properties dialog box, click the Draw tab, and then click Convert to Paint Object.

Instructor displays a warning that the conversion you are about to perform cannot be undone.

3 Click OK to convert the object, or click Cancel to cancel the conversion.

4 *Optional.* Close the Properties dialog box.

Tip If your Instructor application will run on a variety of systems, check the paint object display on several display devices to ensure that the appearance of the converted object is acceptable.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- To convert a picture object to a paint object, send the *convertPicture* message:

```
selection = picture ID 3
send convertPicture
```

After the picture object is converted to a paint object, you can adjust its handles to crop the image.

Creating transparent areas in paint objects

You can create paint objects with transparent areas. When you select the Use fill color as chromakey check box in the Properties dialog box, Instructor uses a paint object's fill color to determine the transparent portion of the graphic. Only the portions of the graphic that match the fill color will become transparent.

For example, let's say that you import a graphic of a red logo against a black background, and you want to place the logo on a white page of your book. Instead of presenting the logo on the black background, you want the logo alone to "float" on the white page. If you set the fill color of the graphic to black, the original black background of the logo graphic will be transparent—that is, it will not be visible—and the logo will appear by itself on your page.

If you plan to export your book to DHTML, all images that use chromakey should be converted to 256 colors before you import them into ToolBook. This will speed up the export process and can produce a better-looking image.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set the chromakey property of a paint object:**

Using the interface



1 Select the paint object, and then click the Properties button on the Instructor toolbar.

2 In the Properties dialog box, click the Draw tab, and then select Use fill color as chromakey.

3 Click Fill color, and then choose the color you want to be transparent in the paint object.

4 *Optional*. Close the Properties dialog box.

Using OpenScript

◆ To set the *chromakey* property of a paint object, send the *useChromakey* and *fillcolor* messages:

```
useChromakey of paintObject "tiger" = true
fillColor of paintObject "tiger" = green
```

You can also set transparent areas in bitmap resources. For more information, see Chapter 17, "Using resources."

Creating and applying a color palette

If your users will run their computers using the 256-color display setting and your application includes many color-intensive images, you may discover some limitations in the way that your graphics are displayed.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Most color display devices are capable of displaying a wide range of colors. However, an 8-bit display device can show only 256 colors at one time, so it maintains a palette with 256 entries to represent the colors it can display. All Windows applications share this palette, called the system palette, to display colors. In a 256-color system, Windows manages requests from applications, such as Instructor, for colors in the system palette. Each application and object can choose a different selection of 256 colors from millions of possible colors. The list of colors needed to display an application or object is its logical palette. For example, an imported picture object in Instructor includes a logical palette that Instructor uses to request colors from the Windows system palette.

A palette is realized when Windows maps the entries in a logical palette to the system palette. When all entries in the system palette are in use, Windows matches any unassigned logical palette entries to the closest available system palette entry. When you switch between pages, Windows realizes the palette for the new window just before you leave the last window, which may cause its colors to change—an effect called palette flash.



Note Users will detect palette flash only on computers set to display 256 colors. You may want to test your application using this setting to see whether palette flash occurs.

The best way to eliminate palette flash is to create a common palette from all of the graphic elements in your book using the Palette Optimizer. Then, apply this palette to all of the graphic elements and assign it as the book palette.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To create a common palette from all of the graphics used in your application:**

1. From the Tools menu, point to Applications, and then choose Palette Optimizer.
2. In the Palette Optimizer utility, on the Create Palette tab, click Browse.
3. Navigate to and select the files from which you want to create the palette, and then click OK to close the Open dialog box.

   The files appear in the Create palette from window.
4. Click Build Palette to create a common palette from all the graphics you have identified.
5. Name and save the palette in the Save Palette dialog box. If you intend to apply the palette to your file list immediately, select Copy to clipboard.

After you have created a common palette, you can apply this palette to all of your graphics using the Palette Optimizer.



Tip You can select more than one file at a time in the Open dialog box by holding down the SHIFT key (to select a series of contiguous files) or the CTRL key (to select noncontiguous files) as you click.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To apply a palette to all of the graphics used in your application:**

1. From the Tools menu, point to Applications, and then choose Palette Optimizer.
2. In the Palette Optimizer utility, click the Apply Palette tab.
3. In the Palette name box, enter the name of the palette you want to apply, or use the Browse button to the right of the field to navigate to and select a palette file. To apply the palette that is currently on the Clipboard, leave the Palette name box empty.
4. To select the graphics to which you want to apply the palette, click the Browse button beneath the Apply palette to list.
5. Navigate to and select the graphic files, and then click OK to close the Open dialog box.

   The files appears in the Apply palette to list.
6. Choose Dither images to palette or Find nearest color in palette.

   Dithering mixes palette colors to achieve a color; finding the nearest color substitutes a palette color for the original image of the color. For example, if the original image contains green and the palette contains only yellow and blue, dithering places yellow and blue pixels side by side to create the illusion of green. By contrast, finding the nearest color might change all of the green in the image to blue.
7. Click Apply Palette.



Note If you change a graphic's palette, you will need to reimport the graphic into Instructor before the change becomes apparent.

After you have created a common palette from the graphics files, applied this palette to all the graphics files, and imported the graphics files into your book, specify this new palette as the book palette using the Properties for Book dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

4 Creating books and pages

5 Working with the Catalog

6 Working with objects

7 Setting object properties

8 Working with text

9 Using graphics

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To assign a book palette:**

1 From the Object menu, choose Properties for Book.

2 On the Draw tab, under Palette, click Choose.

3 In the Choose Palette dialog box, import the palette file (with a .pal file extension) you created, select the palette in the graphical list of resources, and then click OK.

4 *Optional*. Close the Properties for Book dialog box.



For more information about color palettes and 256-color display devices, refer to online Help.

PART III

Adding interactive features to your application

Using powerful objects from the Catalog, you can build dynamic, interactive applications to present your learning material. In this part of the *ToolBook II Instructor User Guide,* you will learn about using buttons; creating a quiz using question objects; hiding, showing, and animating objects; and using multimedia in your application.

IN THIS PART

Chapter 10 Working with buttons and hotwords

Chapter 11 Working with list boxes and combo boxes

Chapter 12 Creating a quiz using question objects

Chapter 13 Hiding, showing, and animating objects

Chapter 14 Using multimedia



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 10

Working with buttons and hotwords

You can add buttons and hotwords to your application when you want your users to interact with your book. A ToolBook II Instructor button, like any Windows button, is an object that the user can click to cause something to happen. A hotword is a selection of text that the user can click like a button. Buttons and hotwords can be used to navigate between pages, as controls for setting options, to start and stop an animation, and for a variety of other uses. This chapter describes how to use buttons and hotwords in your application.

IN THIS CHAPTER

About buttons **246**

Defining how buttons look **246**

Using and modifying buttons **248**

Adding a graphic to a button **258**

About hotwords **263**

Creating and modifying hotwords **264**

Editing hotwords **267**

Removing hotwords **269**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About buttons

Buttons facilitate user interaction. They are commonly used as controls—to turn options on and off, to navigate between pages, or to choose one option from among a related set of options.

For example, a button can:

- be hyperlinked, taking the user to another page, another book, or a Web site.
- start an animation or a multimedia feature.
- allow users to check their responses to questions or score their work.

Defining how buttons look

The Catalog contains a wide variety of buttons that you can use to add interactive behavior to your learning application—from prebuilt navigation objects to graphical question objects.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

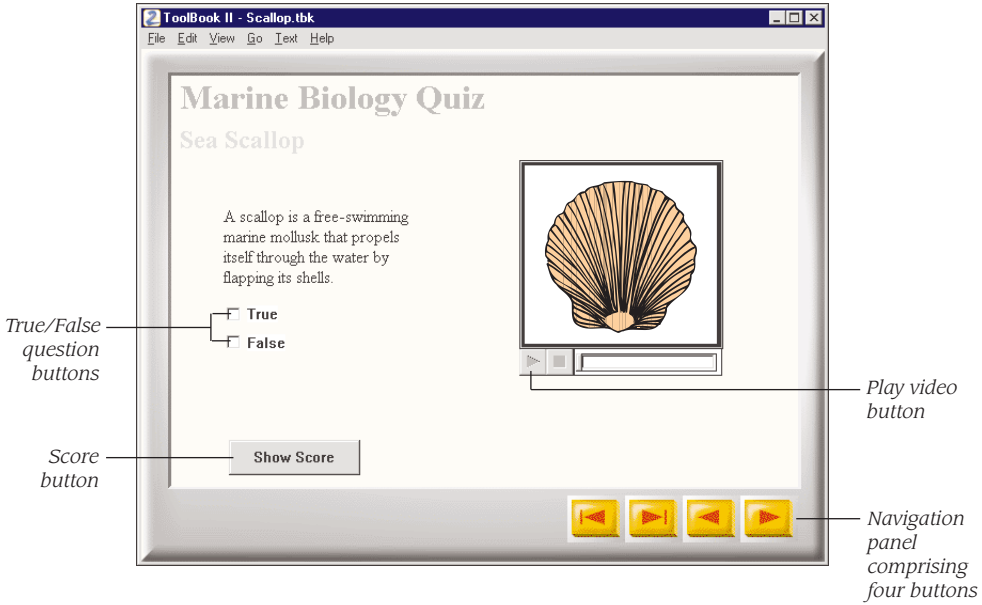


Figure 1
Buttons are used to provide choices and control actions

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Although each button in Figure 1 looks very different from the others, all are button objects; a button's appearance is defined by various property settings that you can specify. For example, a button's border style is a property that determines whether the button appears as a rectangle, a radio button, a check box, or some other type of button. For more information about border styles and other button properties, see "Setting button properties," later in this chapter.

In addition to different border styles, a button also has a state, indicating its active status, which you can represent with a graphic. You can assign a graphic to a button in the Properties for Button dialog box. For details about adding a graphic to a button, see "Adding a graphic to a button," later in this chapter.

Using and modifying buttons

There are three ways to create a button. You can drag a button with the functionality you want from the Catalog to your page. You can also create a button using the Instructor tool palette. Finally, you can use OpenScript to draw a button.

After you have created a button, you can set properties that determine how the button looks—for example, you can format its border style, color, or caption text—using the Properties for Button dialog box.

Buttons that you draw using either the tool palette or OpenScript, as well as the general-use buttons available in the Catalog, do not have predefined behaviors. These buttons can be hyperlinked to other locations or programmed in the Actions Editor to perform a variety of tasks. Alternatively, you can use OpenScript to define any aspect of a button's behavior and appearance.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Adding a button from the Catalog

The Catalog contains a wide variety of basic button objects for general use, as well as buttons with extended properties designed for specific purposes, such as media players, navigation panels, and question objects. To add a button to your book, drag it from the Catalog to a page or background and then define its properties in the Properties for Button dialog box.

➤ **To create a button and set its properties:**



1 Click the Catalog button on the Instructor toolbar to open the Catalog.

2 In the Catalog, explore the categories that contain buttons.

For example, general-use buttons are available from the Buttons category. Buttons with built-in functionality are available from these categories: Action Objects, Scoring and Tracking Objects, Navigation Objects, and Media Players.

3 Drag the button you want from the object pane, and then drop it onto your page.



4 Select the button, and then click the Properties button on the Instructor toolbar.

5 In the Properties for Button dialog box, select or enter the properties you want.

6 *Optional*. Close the Properties dialog box.



Some buttons, such as the buttons in a navigation set, also have extended properties that you can specify. To quickly determine whether a button has extended properties, you can select the button and view the Extended Properties button on the toolbar which becomes available if the button has extended properties.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating a button using the tool palette

There are six tools on the tool palette that you can use to draw Instructor button objects:



- **Pushbutton tool** Use a pushbutton to prompt a user action, such as navigation from one page to another.



- **Label Button tool** Use a label button to provide quick access to the button's functionality without the use of a mouse. The user presses the ALT key in combination with the underlined letter in the button's caption to achieve a button click.



- **Radio Button and Radio Button 3D tools** Use a radio button in a group to allow the user choose one option from a set of mutually exclusive options. When one option in the group is selected, the others are not selected.



- **Check Box and Check Box 3D tools** Use a check box to allow a user to turn options on and off.

➤ **To create a button using the tool palette:**

1. From the tool palette, choose the button tool of your choice.
2. Click on your page and drag the cursor until the button is the desired size.

Creating a button using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To create a button using OpenScript:**

- Use the *draw* command with the *button* object type, specifying the bounds of the button in page units. For convenience, you can assign a name to the button immediately after drawing it. For example, to draw a one-inch-square pushbutton:

```
draw button from 1440,1440 to 2880,2880
name of selection = "Next Page"
```

Setting button properties

You can select a button and then set its properties in the Properties for Button dialog box. Each tab contains properties that set some aspect of a button's behavior and appearance.

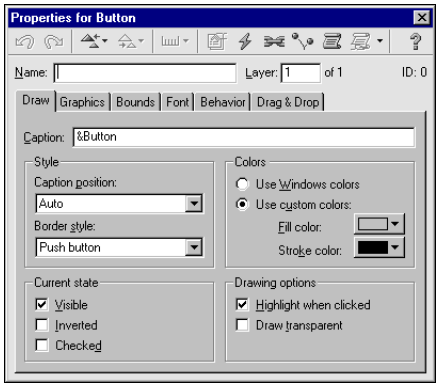
➤ **To set the properties of a button:**

Using the interface



1 Select the button, and then click the Properties button on the Instructor toolbar.

The Properties for Button dialog box appears.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

2 Click one or more tabs and select or enter the options you want to set, as described in the table below.

Instructor button properties

Click this tab...	To do this...
Draw	Set the caption text and position, border style, drawing and highlight options, color options, and current state options
Graphics	Specify a button graphic and set graphic options for each button state
Bounds	Set the button's position and size
Font	Set the font, style, size, effects, and formatting options
Behavior	Specify the button's behavior at run time
Drag & Drop	Specify the appearance and behavior of the button when it is dragged or dropped at Reader level

3 *Optional.* Close the Properties for Button dialog box.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

To specify properties for the button, set the:

- *caption* property to specify the text that appears on the button:

```
caption of button "Next Page" = "Next Page"
```

- *captionPosition* property to specify where the caption appears. If you have a graphic attached to the button, this property sets the caption relative to the graphic; if you don't have a graphic, this property centers the caption:

```
captionPosition of button "Next Page" = left
```

- *highlight* property to see a visual indication when the button is clicked:

```
highlight of button "Next Page" = true
```

- *excludeTab* property to exclude the button from the tab order:

```
excludeTab of button "Next Page" = true
```

- *enabled* property to determine whether the button is dimmed or available (*false* dims the button):

```
enabled of button "Next Page" = false
```

- *borderStyle* property to specify the button's border style:

```
borderStyle of button "Off" = checkBox
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can specify a variety of button border styles using the Properties for Button dialog box or OpenScript.

Button border styles

Example	Border style	OpenScript value
Button	None	*none*
Button	Pushbutton	*pushbutton*
Button	Command	*commandbutton*
Button	Rectangle	*rectangle*
Button	Rounded	*rounded*
Button	Shadowed	*shadowed*
☐ Button	Check Box	*checkBox*
☐ Button	Check Box 3D	*checkBox3D*
○ Button	Radio Button	*radioButton*
○ Button	Radio Button 3D	*radioButton3D*
Button	Label Button	*label*

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Formatting caption text

You can specify various character settings for a button's caption in the Properties for Button dialog box.

➤ **To format a button's caption:**



1 Select the button, and then click the Properties button on the Instructor toolbar.

2 Click the Font tab in the Properties for Button dialog box.

3 Choose the font, style, size, effects, and other options you want.

4 *Optional.* Close the Properties for Button dialog box.



Tip Because text doesn't wrap in a button caption, you might need to resize a button to encompass its caption. For multiple lines of text, use a field instead of a button.

You can also specify an access key for a button, which appears as an underlined character in the button's caption. An access key allows a user to achieve a button click via the keyboard. The user can either click the button with the mouse or press the ALT key in combination with the underlined character in the caption. To set the access key, type an ampersand (&) before the letter you want to use as the access key. For example, to create the caption *Style*, where *e* is the access key, type *Styl&e*.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating hyperlinks

Hyperlinks allow you to connect information in a variety of ways. For example, you can define a hyperlink that navigates to or pops up a related page to display more information about a topic. You can also define special effects for hyperlinks. For example, when the link goes to another page, you can specify a transition effect. When the link pops up another page, you can select a style for the pop-up window. In addition, you can create links that navigate to a URL on the Internet or an intranet.

➤ **To create a hyperlink for a button:**

1. Right-click the button, and then choose Hyperlink from the right-click menu.

   The Hyperlink dialog box appears.

2. In the Type of link list, select the kind of link you want to use.
3. *Optional*. If you want to display the page in a window you've created using a viewer, select the name of the viewer in the In window list.
4. Under Link to, select or enter the page or URL option you want.
5. *Optional*. Do one of the following:
   - Specify a transition effect for the hyperlink. Transition effects are available only when you choose Go to another page as the type of link you want to use.
   - Specify a style for the hyperlink's pop-up window. These settings are available only when you choose Pop up another page as the type of link you want to use.
6. Click OK.

For more information about hyperlinks and linking options, see Chapter 6, "Working with objects."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Scripting a button

You can write a script specifying how a button should behave using the script editor. For example, the following script displays a question in a pop-up dialog box and responds to a user's choice:



Note Keep in mind that you should not write scripts for objects if you plan to export your book to DHTML.

```
to handle buttonClick
   request "Do you want to continue?" with Yes or No
   if it = yes
      go to next page
   end if
end buttonClick
```

Like any other object, buttons can be grouped. You can write a script that sets the behavior of individual buttons and the behavior of the group. For example, you might create several buttons and write a script for each that defines the behavior of a specific tool. You could then group the buttons into a tool palette and write a script to define different aspects of the group's behavior, such as the default location of the palette.

For details about using OpenScript to define button behavior, refer to the *Programming in OpenScript* online book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Programming a button with the Actions Editor

You can use the Actions Editor to easily program the behavior of a button without using OpenScript. In the Actions Editor, you can program a button to play media, prompt users for input, and accomplish a variety of other tasks by linking an action or series of actions to a button click. You can also set conditions for behavior or insert loops that execute the same actions repeatedly. The behavior that you create using the Actions Editor will export to DHTML automatically by using the ToolBook II Web Specialist.

For details about using the Actions Editor to add interactive behavior to your applications, see Chapter 15, “Using the Actions Editor,” and Chapter 16, “Creating action sequences: Practical examples.”

Adding a graphic to a button

Graphics can help convey the purpose and availability of a button to the user. For example, you might use a graphic to indicate that a button is disabled. You can use graphics in combination with a caption, or leave the caption off altogether.

You can assign a graphic to a button state using the Properties for Button dialog box. Alternatively, you can add a graphic to a button by dragging a graphic object from the Catalog to the button. You can also use OpenScript to define this property. You can assign as many as four different graphics to a button (one for each of the four button states).



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Graphics for each button state

Dialog box option	OpenScript property	Description
Normal	*normalGraphic*	Displayed when button is in its normal state
Invert	*invertGraphic*	Displayed when button is being pressed
Disabled	*disabledGraphic*	Displayed when button is disabled
Checked (or Rollover)	*checkedGraphic*	Displayed when button is checked, or displayed as the rollover graphic when the mouse pauses over the button if this option is selected in the Properties for Button dialog box

➤ **To add a graphic to a button:**

Using the interface



1. Select the button to which you want to add a graphic, and then click the Properties button on the Instructor toolbar.
2. In the Properties dialog box, click the Graphics tab.
3. Under Button graphics, choose a button state (Normal, Invert, Disabled, or Checked), and then click Choose Graphic.

   The Choose Graphic dialog box appears.
4. Select the type of resource you want (bitmap, cursor, or icon) in the Available resources list.

   The resources listed in the Choose Graphic dialog box include all the resources of the selected type that have been imported into the current book. Any new resources you import are added to the book's resource list.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

5 Do one of the following:

- To use an existing resource, select the graphic you want in the graphical list of resources.
- To import a new resource, click Import. In the Import Graphic or Import Resource dialog box, locate the graphic file you want and click OK; then, select the resource in the graphical list of resources.

6 Click OK to close the Choose Graphic dialog box.

7 *Optional.* Under Graphic options, do one or more of the following:

- If you want the bitmap to fill the button, select Stretch the graphic to fit the button. (You cannot stretch icons or cursors because they have a fixed size.)
- If you want the button to size itself to the graphic, select Size the button to fit the graphic.
- If you want to use the Checked graphic as the rollover graphic (the graphic you see when the pointer is paused over a button), select Use checked graphic as rollover graphic.

8 Repeat steps 3–7 to assign graphic resources for the other button states.

9 *Optional.* Close the Properties for Button dialog box.

You might need to resize the button to see the graphic.

Using OpenScript

- To apply a graphic to a button, set the *checkedGraphic, disabledGraphic, invertGraphic,* or *normalGraphic* property of a button to a bitmap, cursor, or icon resource. For example:

```
normalGraphic of button "Baseball"=icon "Ball1"
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating transparent areas in a bitmap

You can create transparent areas in paint objects and bitmaps that you have added to your book as a resource with the Use chromakey option. The Use chromakey option appears in the Resource Manager dialog box, on the Graphics tab of the Properties for Button dialog box, and on the Draw tab of the Properties for Background dialog box. Instructor uses the Use chromakey setting in conjunction with a color value to determine the portions of the bitmap that will be transparent.

Note The Use chromakey option is applied to all instances of a resource in your book. For example, if the same bitmap resource is used as a graphic in a button and as an inline graphic in a field, both instances of the bitmap display the transparency.

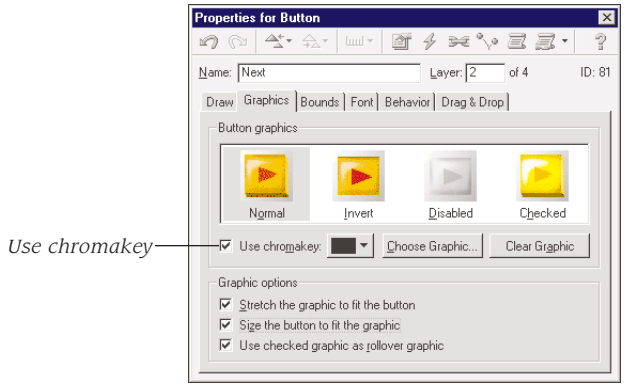


Figure 2
The Use chromakey option

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To create a transparent area in a bitmap:**

1 In the Properties for Button dialog box, click the Graphics tab.

2 Under Button graphics, select the button state (Normal, Invert, Disabled, or Checked) containing the bitmap in which you want to create a transparent area.

For more information about assigning a bitmap to a button state, see the preceding section, “Adding a graphic to a button.”

3 Select Use chromakey, and then click the Color button.

4 In the Color dialog box, select a color, and then click OK.

Portions of the bitmap resource that match this color value become transparent.

Displaying the graphic and caption

You can adjust the layout of a button’s caption in relation to the button graphic. By default, Instructor displays a caption below the graphic in pushbuttons and command buttons, and to the right of the graphic in check boxes and radio buttons. Specify the position of the button graphic in relation to the caption on the Draw tab of the Properties for Button dialog box. If there is no graphic assigned to a button, the position setting has no effect for most button types.

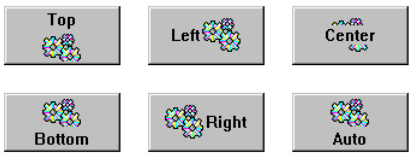


Figure 3
Caption position is relative to the button graphic

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About hotwords

Hotwords are special areas of text that you can create within fields. You can make a hotword from any group of characters in a field. For example, a hotword can be a few characters, a word, a phrase, a sentence, or all the text in a field. Several hotwords can appear in the same sentence, but no two hotwords can overlap. Inline graphics can also be hotwords.



Tip You can create hotwords that play media using a special media player object from the Catalog.

A hotword generates events and sends messages to other objects. Hotwords function much like buttons, except that they are part of the text of a field. At Reader level, a user can click a hotword to navigate or perform some other action. The user can easily identify a hotword because the pointer changes to a hand when it is paused over a hotword.

Hotwords are invaluable for building interactive books capable of complex actions. Instead of using many buttons on a page for all of the options you want to give users, you can make text function as an important part of a book's programmed actions. Hotwords provide another level of individual control in the interface. For example, you can use hotwords to define difficult terminology, saving space and allowing you to create an application appropriate for more than one level of expertise. Experts don't have to search through a lot of definitions, while novices can access the information they need.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

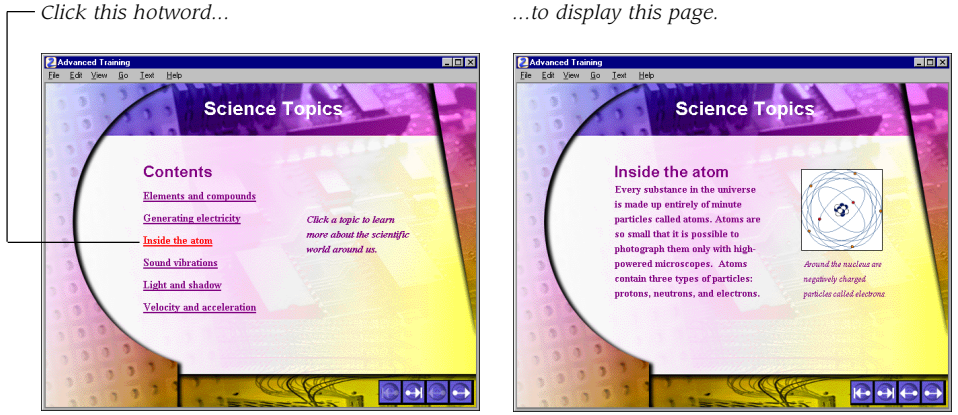


Figure 4
Hotwords provide flexible access to information

Creating and modifying hotwords

You create and modify hotwords at Author level. Like other Instructor objects, hotwords have properties that you can set to define their physical appearance. After you create a hotword, you define a hyperlink for it.

Note In order for Instructor to run a hyperlink, a text field must be set to recognize mouse events. On the Behavior tab of the Properties for Text Field dialog box, the option labeled Lock the text and activate mouse events must be selected.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To create a hotword and define a hyperlink:**

Using the interface

1. Select the text or inline graphic that you want to make into a hotword.
2. Choose Create Hotword from the Text menu, or press CTRL+H.
3. With the hotword still selected, from the Object menu, choose Hyperlink, or press CTRL+SHIFT+H.
4. In the Hyperlink box, specify the type of link and the link destination, and then click OK.
5. Press F3 to go to Reader level, and then click the hotword to test the hyperlink.

Using OpenScript

- Use the *createHotword* message to create a hotword from selected text:

```
Select third word of textline 1 of text of field "Intro"
send createHotword
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set the properties of a hotword:**

Using the interface



1 Select the hotword, and then click the Properties button on the Instructor toolbar.

2 Select one or more options to set the hotword's properties:

- **Style** Specifies whether the hotword appears in color, with a frame around it, with no distinction from the surrounding text, and so on. Choose Book default to have the hotword appear in the style indicated by a book's default hotword style.
- **The hotword is highlighted** Always shows the hotword in its highlighted (pressed) state.
- **Highlight when clicked** Highlights the hotword when the user clicks it.

3 *Optional*. Close the Properties for Hotword dialog box.

A hotword must contain at least one character. You can format hotwords just as you format other text. For details about formatting text, see Chapter 8, "Working with text."

Using OpenScript

- Use the book's *hotwordStyle* property to set all hotwords in a book to a color, add a border, or to have no style attributes. The default value of *hotwordStyle* is *color*:

```
hotwordStyle of this book = color
```

- Use the *hotwordStyle* property of the hotword to set a different style for that hotword:

```
hotwordStyle of hotword "Popup" = frame
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

- To remove a hotword's formatting and set it to the style of the book, set the hotword's style to *bookDefault*:

```
hotwordStyle of hotword "Popup" = bookDefault
```

- The *hotwordColor* property specifies the color value of the hotwords. The default for a book's *hotwordColor* property is red. For example:

```
hotwordColor of this book = cyan
```

- Use the *invert* property to make the hotword appear highlighted:

```
invert of hotword "Popup" = true
```

- Use the highlight property to cause a hotword to flash momentarily when clicked:

```
highlight of hotword "Popup" = true
```

Editing hotwords

You can select and edit a hotword just as you edit other text in Instructor. When you edit hotword text, use these guidelines:

- If the insertion point is within a hotword, any additional characters you type become part of the hotword.
- Text you paste into a hotword becomes part of the hotword.
- If you select a hotword and include text that is not part of the hotword in the selection, Instructor treats the selection as text, not as a hotword.
- You can delete a hotword by using the BACKSPACE key or the DELETE key. As long as one character of the hotword text remains, the hotword's object properties (such as hyperlinks) remain. Instructor discards the hotword as an object when the last character of the hotword is deleted.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

To make hotword editing easier, you can choose Show Hotwords from the View menu to display all the hotwords on a page. Alternatively, send the OpenScript *showHotwords* message.

Applying color to hotwords

Both books and hotwords have a hotword style property that determines how a hotword is distinguished from the surrounding text—with color, for instance. A book's hotword color property (defined on the Draw tab of the Properties for Book dialog box) defines the color value of all the hotwords in the book. However, an individual hotword style property (defined in the Properties for Hotword dialog box) overrides the book's hotword style property.

You can apply a color to a specific hotword that is different from the color applied to other hotwords in a book, or even make different parts of the hotword different colors. Set the hotword's style to None, and then apply formatting by setting the hotword's text color in the Properties for Hotword dialog box.

After you have applied special formatting to an individual hotword, you can easily reapply the book's formatting by changing the hotword's style to Book default.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Removing hotwords

You can remove a hotword by changing it into regular text.

➤ **To change a hotword into regular text:**

Using the interface

1 Place the insertion point in the hotword or select any part of it.

2 From the Text menu, choose Remove Hotword.

Instructor changes the hotword into regular text and discards its script and hyperlink properties.

Using OpenScript

◆ To remove a hotword, send the *removeHotword* message:

```
select third word of textline 1 of text of field "Intro"
send removeHotword
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 11

Working with list boxes and combo boxes

You can display multiple choices for your users with a list box, which is a type of field in ToolBook II Instructor, or with a combo box object. This chapter describes how to create and format list boxes and combo boxes in your application.

IN THIS CHAPTER

About list boxes and combo boxes **272**

Using list boxes or combo boxes on the Internet **274**

Creating list boxes **274**

Setting list box properties **276**

Adding text to a list box **279**

Creating combo boxes **281**

Setting combo box properties **283**

Adding text to combo boxes **284**

Determining user selection **286**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About list boxes and combo boxes

List boxes and combo boxes contain and present lists of options to your users. A list box is a type of field. It organizes choices into a list of items from which the user can make a selection. You determine the size of the list box; if there are more items than can fit in the area you have defined, you can add a scroll bar.

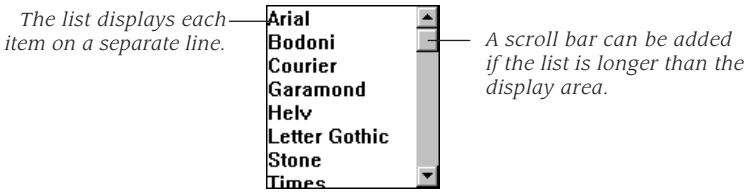


Figure 1
List box with a scroll bar

You can use a list box when:

- the set of choices is too large for radio buttons or check boxes.
- space is tight, making radio buttons impractical.
- you want to give equal weight to all of the list choices.
- you want the choices to be visible to avoid confusion or mistakes.
- all the choices are fixed.

Most of the time, you'll create a list box using a text field rather than a record field. But if you have a similar list on several pages with the same background, you can create a list box from a record field.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

A combo box has three parts: a single-line edit box, a pushbutton, and a drop-down list box. When the user clicks the pushbutton, the drop-down list appears. Users can type in the edit box or select an item from the drop-down list. The edit box displays the currently selected item.

When the user selects an item in the drop-down list, the list box is hidden and the newly selected item appears in the edit box. By default, users cannot enter text in combo boxes at Reader level; that is, the user can only select the items in the drop-down list. You can also create a combo box that can be edited by the user. In this case, the user can click an insertion point in the edit box and type any entry.

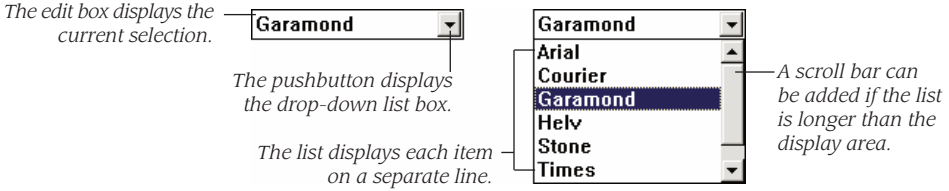


Figure 2
Combo box

You might choose to use a combo box when:

- the set of choices is too large for radio buttons or check boxes.
- space is tight, making radio buttons and list boxes impractical.
- there is a default choice.
- you want users to be able to type in their own choices.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using list boxes or combo boxes on the Internet

You can use list boxes and combo boxes in a native ToolBook II application distributed over the Internet using the ToolBook II Neuron browser plug-in. Additionally, you can use both list boxes and combo boxes in applications that you plan to export to DHTML. For more information about using the Internet to distribute courseware, see Chapter 24, “Distributing applications on the Internet.”

Creating list boxes

There are three ways to create a list box. You can drag a list box from the Catalog to your page. Alternatively, you can create a list box using the list box tool on the Instructor tool palette. Finally, you can use OpenScript to create a list box.

By default, list boxes that you draw using the tool palette are single-select list boxes, in which the user can select only one item at a time. If you want the user to be able to select more than one item at a time, you can change the list box to a multi-select list box using either OpenScript or the Properties for Text Field dialog box. For details, see “Setting list box properties,” later in this chapter.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Adding a list box from the Catalog

The Catalog offers several list boxes that you can drag from the object pane and drop onto your page.

➤ **To add a list box from the Catalog:**



1. Click the Catalog button on the Instructor toolbar to open the Catalog.
2. In the Catalog, select the Text Fields category.
3. Drag a list box object onto your page.

Creating a list box using the tool palette

You draw a list box on your page using the list box field tool.

➤ **To create a list box using the tool palette:**



1. From the tool palette, choose the single-select list box tool.
2. Click in the main window and hold down the mouse button while you drag to draw the list box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating a list box using OpenScript

You can write and execute OpenScript code using the script editor or Command window. List boxes you create using OpenScript will not export to DHTML. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

➤ **To create a list box using OpenScript:**

- To make a field into a list box, set the *fieldType* property to either *singleSelect* or *multiSelect.* For example:

```
draw field from 2000, 1000 to 4000, 4000
name of selection = "Cities"
borderStyle of selection = scrolling
fieldType of selection = singleSelect
```

Setting list box properties

After you have created a list box, you can set properties that determine how the box appears and behaves at Reader level—for example, its border style and color. Because a list box is a type of Instructor field, you set these properties in the Properties for Text Field dialog box. Each tab in this dialog box sets some aspect of the field's appearance or behavior.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

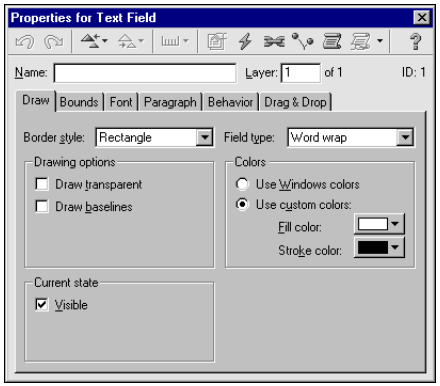
Index

➤ **To set the properties of a list box:**



1 Select the list box, and then click the Properties button on the Instructor toolbar.

The Properties for Text Field dialog box appears.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

2 Click one or more tabs and select or enter the options you want to set, as described in the table below.

Instructor list box properties

Click this tab...	To do this...
Draw	Set drawing options, border style, field type, color, and visibility state options
Bounds	Set the field's position and size
Font	Set the font, style, size, effects, and formatting options
Paragraph	Set text alignment, indentation, spacing, and tabs
Behavior	Specify the field's behavior at run time
Drag & Drop	Specify the appearance and behavior of the field when it is dragged or dropped at Reader level

3 *Optional.* Close the Properties for Text Field dialog box.

For more information about formatting text fields, see Chapter 8, "Working with text."

Modifying field type

A list box organizes field text into a list, with one item per line. By default, list boxes allow only one selection at a time. If you want the user to be able to select more than one item at a time, you can change the single-select list box to a multi-select list box. Instructor highlights the selected list option or options.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

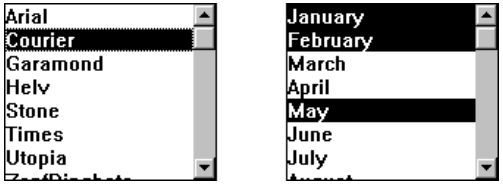
PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



Single-select list box

Multi-select list box

Figure 3
Single-select and multi-select list boxes

➤ **To choose a multi-select field type:**

1 Click the Draw tab of the Properties for Text Field dialog box.

2 In the Field type list, select Multi-select list box.

3 *Optional.* Close the Properties for Text Field dialog box.

Adding text to a list box

You add text to a list box in the same way that you add text to any field: by typing, pasting, or importing. Each text line must be separated from the next item using the ENTER key.

Note Text does not wrap at the ends of lines in a list box. If you want to type in more text than fits on a line, you must first resize the list box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To type text in a list box:

Using the interface

1 Double-click inside the list box.

An insertion point appears in the upper-left corner.

2 Type your text.

Press ENTER after each item in the list.

3 When you are finished, click outside the field.

Using OpenScript

- You enter text in a list box by setting the *text* property of the field. Separate the items in the list with a carriage return/linefeed (CRLF). For example:

```
text of field "Cities" = "Alphaville" & CRLF & "Bellona" \
   & CRLF & "Oz" & CRLF & "Zone"
```

- You can also use *textline* to set each item in the list separately:

```
textline 1 of text of field "Cities" = "Alphaville"
textline 2 of text of field "Cities" = "Bellona"
textline 3 of text of field "Cities" = "Oz"
textline 4 of text of field "Cities" = "Zone"
```

You can also paste text from the Clipboard into a list box.

➤ To paste text into a list box:

1 Copy the text you want to the Clipboard.

2 Double-click inside the list box into which you want to paste the text.

An insertion point appears.

3 From the Edit menu, choose Paste, and then click outside the list box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Note If the text you paste is too long to fit the list box, Instructor displays as much text as possible. You cannot scroll to see missing text, but if you resize the list box, the text becomes visible. You can also set the OpenScript property *borderStyle* to *scrolling*.

For details about working with text in fields, see Chapter 8, "Working with text."

Creating combo boxes

You can drag a combo box from the Catalog to your page, use the combo box tool on the tool palette, or use OpenScript to create a combo box. The size of the area you draw for a combo box determines the width of the edit box. If you create a combo box using the tool palette, the current font and font size determine the height of the combo box's edit box. If you create a combo box using OpenScript, the height is set by the OpenScript *lineCount* property. When you select the combo box, selection handles appear around the edit box portion. You can resize a combo box after you draw it.

Adding a combo box from the Catalog

The Catalog contains a combo box that you can drag from the object pane and drop onto your page.

➤ **To add a comb box from the Catalog:**

1 Click the Catalog button on the Instructor toolbar to open the Catalog.

2 In the Catalog, select the Text Fields category.

3 Drag a combo box onto your page.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating a combo box using the tool palette

You can draw a combo box on your page using the combo box tool.

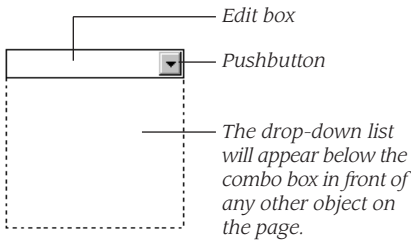
➤ **To create a combo box using the tool palette:**



1 From the tool palette, choose the combo box tool.

2 Click in the main window and hold down the mouse button while you drag to draw the combo box.

The box you draw will include both the edit box and the pushbutton.



Creating a combo box using OpenScript

Combo boxes you create using OpenScript will not export to DHTML. You can use a combo box from the Catalog if you want it to export to DHTML, or draw one using the tool palette.

➤ **To create a combo box using OpenScript:**

◆ Use the *draw* command to draw the combo box and the *lineCount* property to set the height of the drop-down list. For example:

```
draw comboBox from 100,1000 to 5000,2000
name of selection = "myList"
lineCount of comboBox "myList" = 3
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Setting combo box properties

You set combo box properties using the Properties for Combo Box dialog box. Using this dialog box, you can specify properties that determine how the combo box appears and behaves at Reader level.

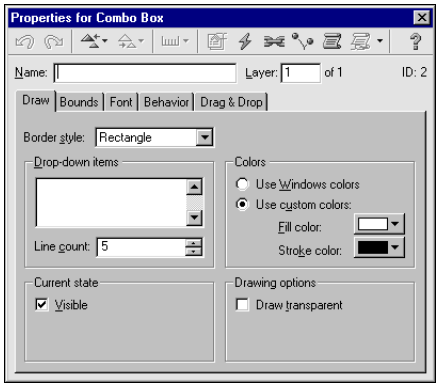
➤ **To set the properties of a combo box:**

Using the interface



1 Select the combo box, and then click the Properties button on the Instructor toolbar.

The Properties for Combo Box dialog box appears.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

2 To specify the behavior of the combo box at Reader level, click the Behavior tab, and then select one or more of the following options:

- **Enable the combo box** Specifies whether the combo box can receive the focus and mouse-event messages at Reader level. Combo boxes are enabled by default.
- **Make the combo box editable** Allows a user to edit the combo box at Reader level.
- **Give the drop-down list a scroll bar if necessary** Adds a scroll bar to the drop-down list if there are more items than will fit.
- **Sort the drop-down list alphabetically** Sorts items alphabetically at Reader level.

3 Click one or more of the other tabs and select or enter the options you want to set.

4 *Optional*. Close the Properties for Combo Box dialog box.

Using OpenScript

- You can use the *scrollable, sortItems, editable, lineCount,* and *enabled* properties to format your combo box:

```
scrollable of comboBox "Long List" = true
sortItems of comboBox "Long List" = true
editable of comboBox "Long List" = false
enabled of comboBox "Long List" = true
lineCount of comboBox "Long List" = 8
```

Adding text to combo boxes

You can add text to a combo box by typing in the field or by using OpenScript.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To enter text in a combo box:

Using the interface

1. Select the combo box you have added to the page.
2. To enter default text that will appear in the edit box, double-click inside the edit box to display an insertion point, and then type your text.
3. Click the pushbutton to display the drop-down list.

   Instructor places the insertion point in the drop-down list.
4. Type as many items as you want, pressing ENTER after each item.

   If you want to change the character formatting of the text in the drop-down list, select the list, choose Character from the Text menu, and then enter your changes in the Character dialog box.
5. When you have finished, click outside the combo box.

   The drop-down list box closes.

Using OpenScript

- Use the *dropDownItems* property to add items to the drop-down list. If you want default text to appear in the edit box, use the *text* property:

```
draw comboBox from 100,1000 to 5000,2000
lineCount of comboBox "myItems" = 3
dropDownItems of comboBox "myItems" = "item 1" & CRLF & "item 2" \
   & CRLF & "item 3"
text of comboBox "myItems" = "item 1"
```

LineCount controls the number of lines in the drop-down list, while *dropDownItems* controls the text displayed in the list box. These properties don't have to match. If the number of *textlines* in *dropDownItems* is greater than *lineCount* (if you specify more list options than will fit in the visible drop-down list), Instructor displays as much of the list as it can. Add a scroll bar so that the user can see all the text.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Determining user selection

You can use list boxes and combo boxes to customize the behavior or content of your application for individual users. For example, you can show a video or direct users to a particular page based on their list box or combo box selection.

To determine and handle a user's list box or combo box selection, you can either build an action sequence in the Actions Editor or write a script in OpenScript. The action sequences that you create in the Actions Editor export to DHTML for use on the Web.

Determining list box or combo box selections using an action sequence

In an action sequence, you determine a user's selection in a single-select list box or combo box using the On select event. The On select event is generated when a user selects a different item from the list. Figure 4 shows the Action Editor with an example of an action sequence that carries out specific actions based on a user's list box selection.

The *selectedItemText* property of list boxes and combo boxes allows you to determine the text of a selected item. You can set this property to select the item in a list box which matches the text you specify. For a combo box, assigning text to the *selectedItemText* property causes this text to display in the combo box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

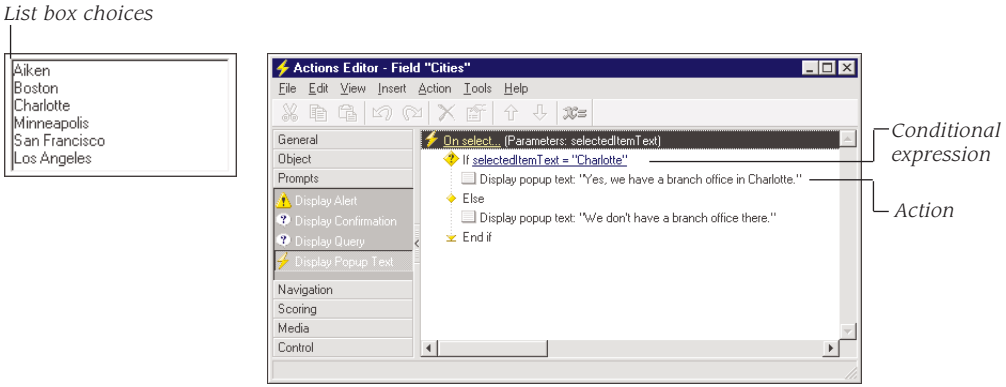


Figure 4
Using an action sequence to determine the user selection

Determining list box selection using OpenScript

The *selectedTextlines* property of the field contains the number of the selected text line, so you can determine the text that has been highlighted in the list box. Use the following script in a button's script (or another object other than the list box) to determine the item that the user has selected in the single-select list box:

```
to handle buttonClick
   --Make sure a line is selected in the single-select list box
   choice = selectedTextlines of field "list"
   if choice < > null
      --Gets the text of the selected list
      vHighlightedText = textline choice of text of field "list"
      request vhighlightedText
   end if
end buttonClick
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

If the field is a multi-select list box, *selectedTextlines* contains a list of numbers for the lines that are highlighted. You can step through the list to pick out individual line numbers and then use the line numbers one by one to extract the highlighted text. For example:

```
to handle buttonClick
   --Determines which line is selected in the list box
   choices = selectedTextlines of field "list"
   if choices <> 0
      step ctr from 1 to itemCount(Choices)
         --Gets the line number of the next selected item
         nextLineNo = item ctr of choices
         --Displays the text of the selected text
         textFromList= textline nextLineNo of text of field "list"
         request "next choice is" && textFromList
      end step
   end if
end buttonClick
```

Determining combo box selection using OpenScript

After the user clicks the pushbutton of a combo box and makes a selection from the drop-down list, the message *selectChange* is sent to the combo box, along with a parameter indicating the user's choice. Use a handler like the following to detect what the user has selected:

```
to handle selectChange itemString
   request "User has selected" && itemString
end selectChange
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

When the user selects an item, the current value of the edit box (the contents of the text property of the combo box) is discarded and replaced with the user's choice. If you want to preserve the old value, save it before the *selectChange* message is sent. One option is to store the old value as a user property of the combo box when the user presses the combo box's pushbutton and the *enterDropDown* message is sent. For example, put both of these handlers in the script of a combo box:

```
to handle enterDropDown
   saveText of self = text of self
end enterDropDown

--Adds the old value of the edit box to the drop-down list
to handle selectChange
   vText = dropDownItems of self
   if (saveText of self & CRLF) is not in (vText & CRLF) then
      --If value is not already in drop-down list, adds it
      dropDownItems of self = vText & CRLF & saveText of self
   end
end
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can also use the *selectedItem* property to determine the selected item by position number in the combo box. Use this script in the combo box to determine what the user has selected:

```
to handle enterDropDown
   --Creates a user property to save value of initial selected item
   my oldSelectedItem = my selectedItem
end

to handle leaveDropDown
   --oldSelectedItem is a user property of the combo box
   --and is set in an enterDropDown handler
   conditions
      when my selectedItem > my oldSelectedItem
         send greaterItemChosen
      when my selectedItem < my oldSelectedItem
         send lesserItemChosen
      else
         send sameItemChosen
   end conditions
end leaveDropDown
```



Note Be sure to handle the three new messages generated by this script, or an error will occur. For more information on handling messages, refer to the *Programming in OpenScript* online book.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 12

Creating a quiz using question objects

An important part of interactive courseware is the ability to evaluate how well a user is learning the material presented. To assist you with this task, ToolBook II Instructor provides question objects that you can use to test student knowledge. This chapter describes question objects and shows you how to use and modify them to suit your application.

IN THIS CHAPTER

About question objects **292**

Working with question objects **293**

Question objects in action **297**

Scoring questions **302**

Specifying feedback **304**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About question objects

Question objects allow you to construct different types of questions, including multiple-choice questions, true/false questions, fill-in-the-blank questions, match-item questions, and so on. Question objects also manage the tasks of determining user responses, scoring responses, and providing feedback.

There are many types of question objects available from the Catalog. You can customize their appearance and behavior or use them as they are provided and simply add your own content.

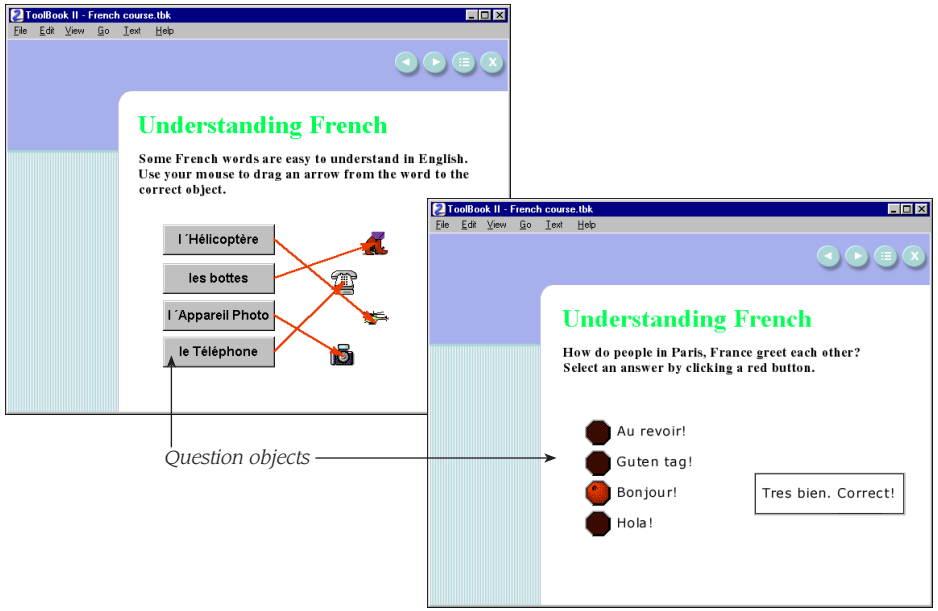


Figure 1
Examples of question objects

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with question objects

You can browse through the question objects in the Catalog, review their descriptions, and then select a specific type of question to use. When you have found the object you want, drag it from the Catalog and drop it onto a page of your book.



Figure 2
Question objects in the Catalog

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To add a question object to a page:



1. If the Catalog is not already open, click the Catalog button on the Instructor toolbar.
2. In the Catalog, select one of the Question categories.
3. Drag the question object you want from the object pane, and then drop it onto your page.

Setting the properties of a question object

After you add a question object to a page, you can specify its appearance and behavior in the Properties for Question dialog box.

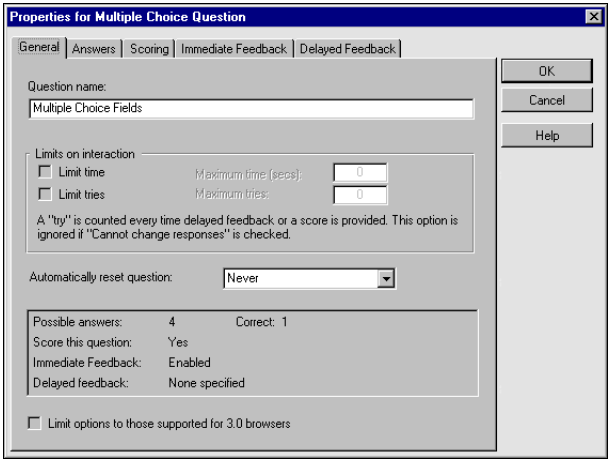


Figure 3
The Properties for Question dialog box

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

The Properties for Question dialog box differs slightly for each type of question object, but most question objects allow you to specify:

- each possible answer to be recognized.
- which answer is correct.
- how many times the student can try to answer the question.
- the options for assigning a score to the question.
- the options for providing feedback to the student.

➤ **To set the extended properties of a question object:**

1 Select the question object.



2 Click the Extended Properties button on the Instructor toolbar.

The Properties for Question dialog box appears.

This dialog box displays information specific to the selected question object. The remainder of this procedure outlines the most common options.

3 Click the General tab and do the following:

- Type a unique name in the Question name box.
- Specify any other options you want.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

4 Click the Answers tab and do the following:

- Under Possible answers, enter the possible answers, specify the correct answer or answers, and then specify any other options you want.

5 Click the Scoring tab and do the following:

- Select Score this question, specify the weight of an answer, and then specify the scoring options you want.

6 Click the Immediate Feedback tab, and then specify the feedback that occurs at the time a user responds to a question.

7 Click the Delayed Feedback tab, and then specify the feedback that occurs at a later (delayed) time.

8 To accept the current property settings, click OK.



For information about the extended properties of a specific question object, refer to online Help.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Question objects in action

The Catalog contains a rich variety of question objects that help you evaluate interactive learning in your application. Several of these are introduced below.

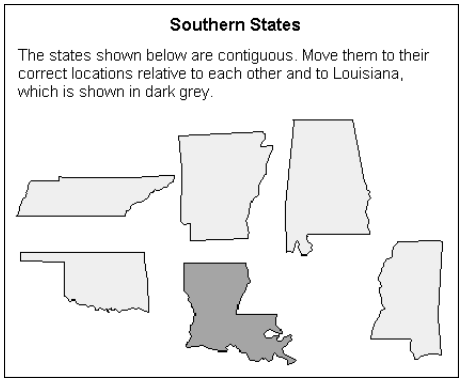


Figure 4
A question created using an arrange-objects object

An arrange-objects object allows you to specify the proper positions of objects on a page and test the student's knowledge of the positions. When the book runs, the objects are scrambled on the page, and the student attempts to drag them back into the arrangement you defined. A student's score is based on the number of objects that are positioned correctly.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

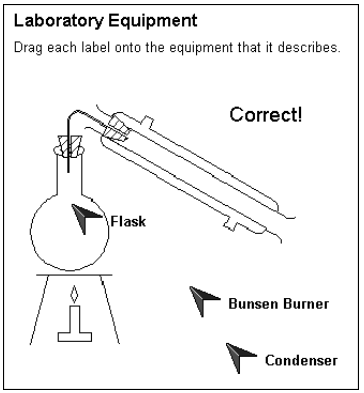


Figure 5
A question created using a drag-objects question

A drag-objects question allows you to define objects as potential answers and test the student's ability to identify the correct answer. The student chooses an answer by dragging objects to one of the potential answers that you have defined. In the example above, a student drags text labels to identify parts.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

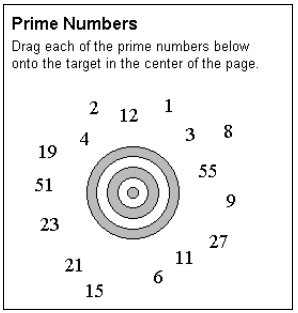


Figure 6
A question created using a drop-target object

A drop-target object allows you to define objects as potential answers and test the student's ability to identify the correct answers. The student chooses an answer by dragging an answer object to the drop-target object. The drop-target object itself is typically a picture of a container or target. Use this question object when the correct answer allows the student to drag multiple objects to a single target.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



Figure 7
A question created using a fill-in-the-blank object

A fill-in-the-blank object allows you to define a word or phrase as the correct response to a question. The student attempts to answer the question by typing a response in the object's response area.

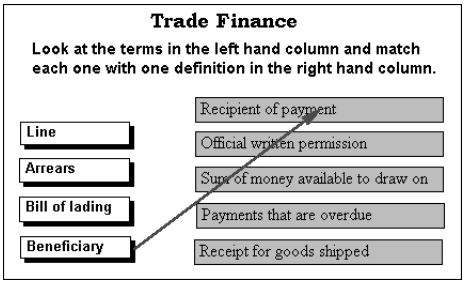


Figure 8
A question created using a match-items object

A match-items object allows you to define pairs of matching items as correct answers and test the student's ability to identify the pairs correctly. The student attempts to match the items correctly by dragging an arrow from each response object to its match.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

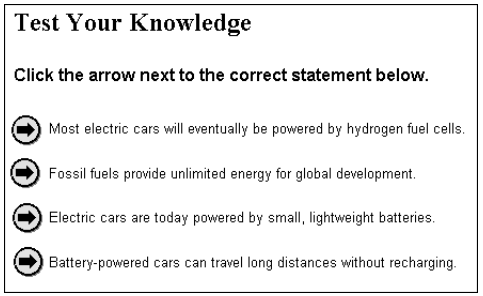


Figure 9
A question created using a multiple-choice object

A multiple-choice object allows you to define a button, field, or other object as a correct answer. The student attempts to answer the question by clicking the button, field, or object. Using this object, you can also designate multiple correct answers and specify randomization (the correct answer appears in a different location each time).

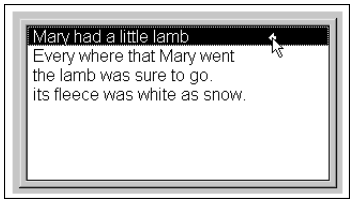


Figure 10
A question created using an order-text object

An order-text object allows you to define an ordered list of words or phrases as the correct response to a question and test the student's ability to reconstruct the list in the proper order. The student attempts to answer the question by dragging the phrases to the correct position.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Scoring questions

In addition to indicating which answer is correct, you can specify that a question object calculate a student's score. You can report the score to the student at the end of the book or record the score in a log file. A log file is a text file that includes information such as the student's score, response to each question, and time spent working on the page. You specify options for a log file in the Properties for Lesson dialog box, which is available from the Object menu. You can also have Instructor report the score to Ingenium from click2learn.com,inc., which can maintain the student's score in a database for administrators to examine later.

At its simplest, scoring is a tally of the questions the student has answered correctly. For more flexibility, you can assign different point values to different questions. You can also assign a weight to each correct answer by specifying a correctness factor within any question object. You can subtract points from the student's score for incorrect answers, and you can limit the number of times a user attempts to answer the question.

➤ **To enable scoring:**

1 Select the question object.



2 Click the Extended Properties button on the Instructor toolbar.

3 In the Properties for Question dialog box, click the Scoring tab.

4 Select the Score this question check box.

5 If necessary, change the default values in the Lowest possible score and Highest possible score boxes.

6 Under Possible answers, select each response and verify that the weight assigned to the answer by the Properties dialog box is correct.

7 Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You display a student's score in your application using the Show Score, Score Page, and Score Quiz buttons available from the Scoring and Tracking category in the Catalog. These buttons are preprogrammed to calculate the score for an individual question object or for several question objects. You can use a scoring button in combination with a score field to display the score to the user. Score fields are also available in the Evaluation category of the Catalog.

Assigning weight to an answer

With most types of question objects, you can assign a weight to every answer. The weight is a percentage of the possible points available for the question. For example, with a fill-in-the-blank question, you might assign different weights to various possible answers. You can also let the Properties for Question dialog box assign the weights, depending on the type of question and the number of answers.

By default, the weight of a correct answer is 100 percent if the question has a single answer, and a fraction of 100 percent if there are several answers. The weight is automatically updated as you mark answers correct or incorrect.

You can modify the weights in the Properties for Question dialog box. You can assign a negative weight to an incorrect answer, making the incorrect answer penalize the score for the question until it is either zero or the minimum score for the question, whichever is lower. For example, with a multiple-choice/multiple-correct question, incorrect answers may have a negative weight that cancels correct answers. The Automatic check box on the Scoring tab makes this negative weight assignment automatic.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

For question objects that can recognize only a single answer at a time, such as a fill-in-the-blank or rating-by-multiple-choice question object, you can assign a higher weight to the most desirable answer and a lower weight to the other answers. For example, if your application solicited an opinion with a question such as "Please rate the service at our hotel," you might use a rating-by-multiple-choice object with three possible answers, weighted as shown below.

Rating by multiple choice

Answer	Weight
Excellent	100%
Average	50%
Poor	0%

Specifying feedback

With question objects, you can specify feedback, which is how the object responds when the student interacts with it. For example, when a student chooses the correct answer for a multiple-choice question, the question object can display the message "Correct."

Providing feedback to a student allows for a higher level of interactivity, which results in better retention. Feedback is traditionally used in computer-based training to provide responses or remediation to students according to how well they answer a question. However, you can use feedback for a variety of purposes, such as guiding a student through a task or through your application.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Types of feedback

Instructor provides several types of feedback that you can specify for question objects. For example, you can:

- play a multimedia clip or sound effect.
- display text.
- navigate to another page in the same book or in another book.
- navigate to a URL (a Web address).

You can combine these methods and use more than one type of feedback in response to the user's choice. For example, you can both play a sound file and display text in a pop-up window when the user selects the correct choice in a multiple-choice question. If you specify more than one feedback option, they are executed in the order in which they appear in the bullet list above. In the example given, the student would hear the sound and then see the text in the pop-up window.

By default, text feedback and video feedback appear in pop-up windows. To designate a specific location on the page for the feedback to appear, use a feedback field, a video player, or a stage from the Catalog. If a feedback field is on the page, Instructor will display feedback text in the feedback field by default. If a video player or stage is on the page, Instructor can display video feedback in the video player or stage if you select the player or stage in the Choose Media Clip dialog box (available when you click the Select Clip button on the Immediate Feedback or Delayed Feedback tab of the Properties for Question dialog box).



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Choosing immediate or delayed feedback

Feedback for a question object can be immediate or delayed. Immediate feedback appears in response to a specific question object; for example, a student clicks a button in a multiple-choice question and hears a beep or sees a pop-up window that says "Try again." Delayed feedback is triggered when the question is scored or when a message is sent to the question object to make it play the feedback. Delayed feedback is useful for providing feedback on multiple-choice/multiple-correct questions where you want a student to select several correct answers and then click a button to receive feedback on the selected set of answers.

If you are creating a scored quiz, do not use immediate feedback to tell the user whether an answer attempt is correct or incorrect. Instructor does not count an attempt that displays immediate feedback as an actual attempt. A question with immediate feedback will be scored as though the student selected the correct answer on the first attempt, even if it took the student three tries to select the correct answer.

Use immediate feedback when you:

- do not plan to score a question.
- want to display feedback (such as "Correct" or "Try again") for each possible answer as the student is answering the question.

Use delayed feedback when you:

- plan to score a question as part of a quiz.
- want to display feedback after the student has finished answering the question.
- want to display different feedback for each of the following outcomes: All correct, All incorrect, or Partially correct.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can sometimes combine immediate and delayed feedback in the same question. A multiple-choice/multiple-correct question can have several active answers. By using immediate feedback, you can have the question object provide feedback for each possible answer as soon as the student selects it, whether the question takes a single answer or multiple answers. When the student is finished with a multiple-choice/multiple-correct question object, you can use delayed feedback to provide feedback for the whole question. For example, you might provide one type of feedback if the student answered the question partially correctly, and a different type of feedback if the student answered incorrectly. You can define both types of feedback for any question object except an order-text object, for which you can define only delayed feedback.

Specifying feedback for a question object

You specify feedback for a question object using the Properties for Question dialog box, which has separate tabs for immediate and delayed feedback.

➤ **To specify immediate feedback for a question object:**



1. Select the question object, and then click the Extended Properties button on the Instructor toolbar.
2. Click the Immediate Feedback tab.
3. Select the Enable feedback at time of response check box.
4. Under Possible answers, select the answer or answers for which you want to provide feedback. You can click the Select All Correct or Select All Incorrect buttons to select all the correct or incorrect answers at one time. (The Select All Correct and Select All Incorrect buttons are available only for question objects of the appropriate type.)

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

5 Specify one or more of the following feedback types (the object activates the feedback in this order):

- **Play media clip** Click the Select Clip button to specify a wave audio (WAV) file or media clip for Instructor to play.
- **Feedback text** Click the Edit Text button to specify text for Instructor to display in a pop-up window or in a text or record field named Feedback that appears on the current page or background.
- **Send message** Click the Edit Message button to specify an OpenScript message that Instructor sends as feedback. These messages are described in the table on the following page.

OpenScript feedback messages

This message...	Does this...
ASYM_PlayFeedback	Causes question feedback to play. If delayed feedback is assigned, it will play; otherwise, immediate feedback will play.
ASYM_PlaySummaryFeedback	Causes delayed feedback to play.
ASYM_Reset	Resets the question to the default state (appearance, score, and so forth).
ASYM_ShowYourScore	Shows the current score for the question.

- **Go to page** Click the Select Page button to specify a hyperlink, indicating a target page in the current book or in another book to which the object should jump.

6 Repeat steps 4 and 5 for each answer.

7 Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To specify delayed feedback for a question object:**



1 Select the question object, and then click the Extended Properties button on the Instructor toolbar.

2 Click the Delayed Feedback tab.

3 Under Feedback to execute on request when the responses are, specify whether the delayed feedback should play when responses are All correct, All incorrect, or Partially correct.

4 Specify one or more of the feedback types for each possible outcome.

For details, see the feedback type descriptions in the preceding procedure.

5 Click OK.

You can display delayed feedback using a preprogrammed Catalog object.

➤ **To display delayed feedback for a question object:**

1 Navigate to the page that contains the question for which you want to display delayed feedback.



2 If the Catalog is not already open, click the Catalog button on the Instructor toolbar.

3 In the Scoring and Tracking category, select the Show Delayed Feedback object.



4 Drag the Show Delayed Feedback object from the object pane, and then drop it onto your page. With the object selected, click the Extended Properties button on the Instructor toolbar.

The Properties for Play Feedback dialog box appears.

5 In the Object to score list, select the name of the question object for which you want to display delayed feedback.

6 Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the Actions Editor to specify feedback

You can use the Actions Editor to specify immediate and delayed feedback for a question. Two question events are available in the Actions Editor: On question answered and On question scored. The On question answered event corresponds to immediate feedback, and the On question scored event corresponds to delayed feedback. These events allow you to define feedback behavior that can be more complex than the behavior available on the Feedback tabs in the Properties for Question dialog box. For example, you could implement an action sequence that stores the number of attempts a student makes before choosing the correct answer to a question, and then display text to the student that shows the number of attempts made. Action sequences that you create to respond to question events will work in conjunction with the settings in the Properties for Question dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 13

Hiding, showing, and animating objects

You can hide and show objects and animate objects to add visual effects to your book. Hiding and showing objects is a simple way to add interactivity, and animation can effectively demonstrate change over time or illustrate a chain of events. This chapter describes how to hide, show, and animate objects using the ToolBook II Animation Editor, the Actions Editor, and OpenScript.

IN THIS CHAPTER

Creating visual effects **312**

Hiding and showing objects **312**

Animating Instructor objects **318**

Playing an animation at Reader level **322**

Modifying an animation in the Animation Editor **327**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating visual effects

Visual effects—such as hiding, showing, and animating objects—can help make your ToolBook II Instructor application dynamic and engaging. To create these effects, you can use:

- the ToolBook II Animation Editor to create path-based or cel-based animations for objects.
- the Actions Editor to create action sequences that hide, show, or animate objects in response to specific conditions.
- OpenScript to change the position, size, or shape of objects, or to display a changing set of pictures that gives the illusion of motion. For a detailed description of how to use OpenScript to create special effects in an application, refer to the *Programming in OpenScript* online book.

Hiding and showing objects

Hiding and showing objects is a simple and effective way to create interactivity in your application. An object hidden at Reader level can be made to show when the user performs a certain action. As shown in Figure 1, for example, a user can click one button to display the assembly of a car and another to display its disassembly. These buttons were designed to hide or show individual car parts.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

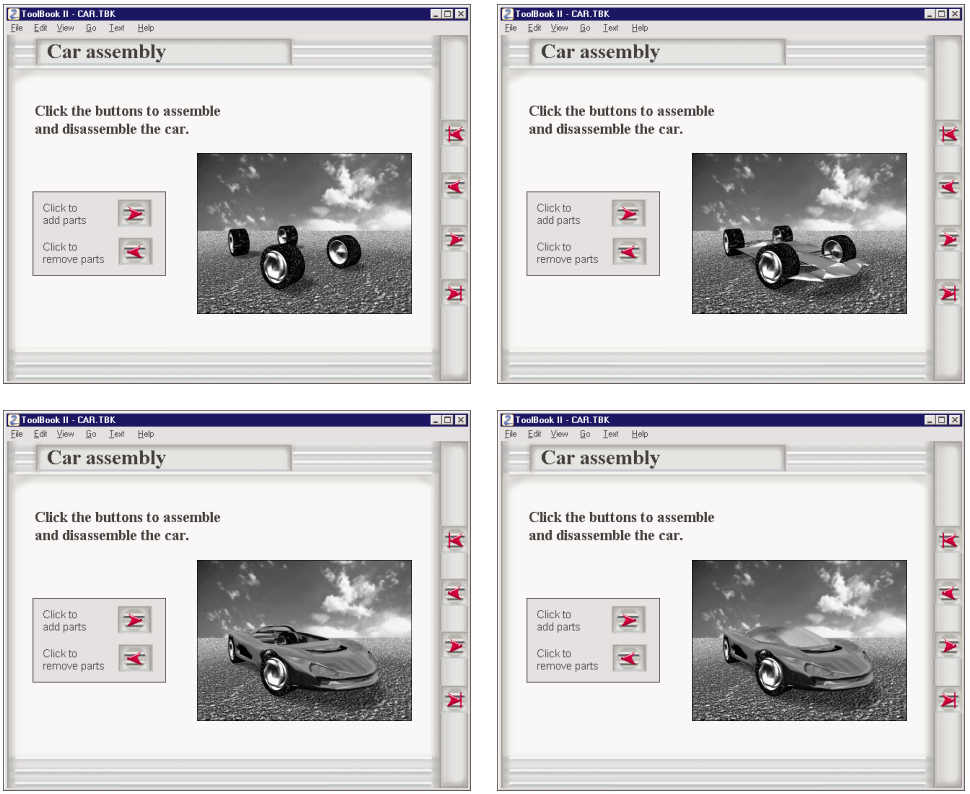


Figure 1
Hide-and-show techniques can illustrate assembly and disassembly

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

To create hide-and-show behavior, you can use an action trigger. Action triggers are available in the Catalog.

You can also use the Actions Editor to create an action sequence to hide and show objects. One advantage of hiding and showing objects in an action sequence is that you can closely control how the effect fits into your overall lesson design. For example, you can hide or show an object as part of a longer sequence of actions, all activated by the same event. You can also create conditional expressions and loops that determine when an object is hidden or shown.

Using an action trigger to hide and show objects

You can use an action trigger to hide and show objects. Simply drag the action trigger from the Catalog onto your page and specify the object to hide or show. You can add a delay when hiding or showing an object using the action-delay trigger.

The following procedure uses an ellipse as an example. You can choose any Catalog object to hide and show.

➤ **To create hide-and-show sequence using action triggers:**



1. Click the Catalog button on the Instructor toolbar to open the Catalog.
2. In the Catalog, click the Draw Objects category. Drag an ellipse object onto your page.
3. From the Action Objects category, drag two action triggers onto your page. Arrange the three objects on your page so that none are overlapping.



4. Select one of the action triggers, and then click the Extended Properties button on the Instructor toolbar.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

5 In the Properties for Action Trigger dialog box, under Objects to trigger, click Add. In the Add an Object dialog box, do the following:

A In the Select object list, select the ellipse.

B Under Perform what action, choose Hide.

C Click OK.

6 In the Properties for Action Trigger dialog box, under Trigger on what event, do the following:

A Select the When clicked check box, and clear the When entering page check box.

B Select the Button is visible at reader check box, and clear the Loop continuously check box.

C Click OK.



7 Select the second action trigger, and then click the Extended Properties button on the Instructor toolbar.

8 In the Properties for Action Trigger dialog box, under Objects to trigger, click Add. In the Add an Object dialog box, do the following:

A In the Select object list, select the ellipse.

B Under Perform what action, choose Show.

C Click OK.

9 In the Properties for Action Trigger dialog box, under Trigger on what event, do the following:

A Select the When clicked check box, and clear the When entering page check box.

B Select the Button is visible at reader check box, and clear the Loop continuously check box.

C Click OK.

10 Switch to Reader level (press F3) to test your hide-and-show sequence.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the Actions Editor to hide and show objects

Using the Actions Editor visual programming tool, you can program one object to hide or show itself or another object.

➤ **To hide or show an object using the Actions Editor:**

1. Select an object (for example, a button or graphic).
2. From the Object menu, choose Actions.

   The Actions Editor appears.
3. In the Event list, select an event that activates the action sequence (for example, On click or On load page).
4. From the Insert menu in the Actions Editor, point to Action, point to Object, and then choose Set Property.
5. In the Properties for Action dialog box, do the following:
   - **A** In the Select the object for which to set the property list, select the object you want to hide or show; or, drag the Choose object button over the object to select it.
   - **B** In the Select the property to set list, select Visible.
   - **C** Under Property value, check Specify the new value for the property as a literal value.
   - **D** In the Select logical value list, select false to hide the object or true to show the object.
6. Click OK.
7. From the File menu in the Actions Editor, choose Update Actions & Close.
8. Switch to Reader level (press F3) to test your sequence.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

For more information about using the Actions Editor to create action sequences, see Chapter 15, "Using the Actions Editor," and Chapter 16, "Creating action sequences: Practical examples."

Using OpenScript to hide and show objects

Using the Command window or the script editor, you can write scripts to show and hide objects. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

- You can show and hide a series of objects in a stack, each of which is a slight variation of the same image:

```
to handle buttonClick
   step ctr from 1 to 6
      if ctr < 6 then
         show ellipse ("ellipse" & (ctr + 1))        --Shows next view
      end if
      hide ellipse ("ellipse" & ctr)                 --Hides the current one
      pause 20
   end step
   show ellipse ("ellipse1")                         --Restores first view
end buttonClick
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Animating Instructor objects

You can animate any object (with the exception of pages, backgrounds, and books). Animation can add visual impact to your learning application and illustrate how things move or change over time.

You can create two types of animations with the ToolBook II Animation Editor:

- **Path-based animation** This type of animation allows you to define the path an object will follow as it moves in an animation. You can use a path animation to simulate or show movement and action, such as a banner that flies across the page.
- **Cel-based animation** This type of animation allows you to define individual views of an object, called cels, that are shown in rapid succession. You can use cel-based animation to simulate motion, such as a globe that spins.

Using the ToolBook II Animation Editor

You use the ToolBook II Animation Editor to create, edit, and manage the animations in a book. When you open the Animation Editor, Instructor copies all of the objects on the page (or background) to the Animation Editor. You use the buttons and slider along the bottom of the window to create, edit, and run animations.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

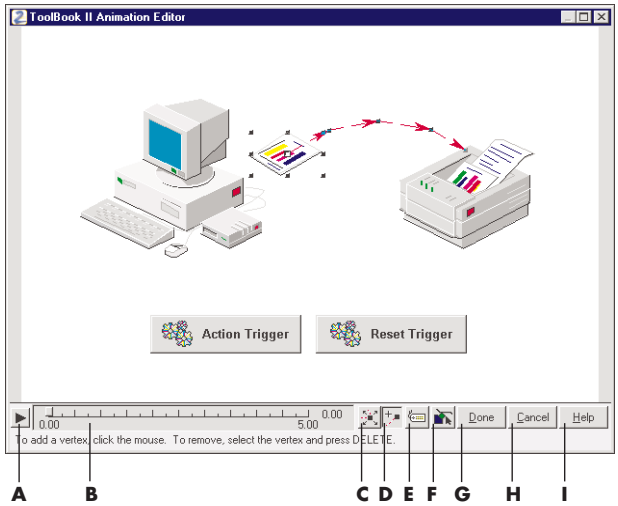
PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



A *Plays the current animation*
B *Allows you to move an object to specific points on the animation path*
C *Allows you to move segments, vertices, or the entire path*
D *Allows you to add new segment. to the animation*
E *Opens the Animation Settings dialog box*
F *Opens the Select Object dialog box*
G *Saves the current animation and closes the ToolBook II Animation Editor*
H *Cancels changes to the current animation and closes the ToolBook II Animation Editor*
I *Opens online Help*

Figure 2
ToolBook II Animation Editor

➤ To open the Animation Editor:

1 *Optional.* In the main window, select the object you want to animate.

2 From the Object menu, choose Path Animation.

To create a path animation in the Animation Editor, you use the mouse to draw the path you want the object to follow. A path consists of segments connected by vertices. You can create a path animation that moves an object while the object changes size. In addition, you can create multiple animations for an object that you can trigger using an action trigger from the Catalog, using a command you create in the Actions Editor, or using OpenScript. When you are finished drawing the path, Instructor compiles information about the path and stores it as a property of the object.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating a path animation

You create a path animation for a particular object by opening the Animation Editor and drawing the path with the Vertex tool. Each click of the mouse adds a new segment to the path.

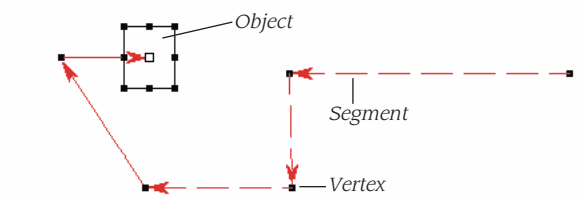


Figure 3
Creating a path animation

➤ **To create a path animation:**

1 In the main window, select the object you want to animate.

2 From the Object menu, choose Path Animation.

Instructor opens the Animation Editor and displays the objects on the current page with the first segment of the animation path for the selected object.



3 *Optional*. To change the location of the first segment, move the cursor over the selected object until the Move cursor appears, and then drag the segment to a new position.



4 To add a segment, click the Vertex button or press the SPACEBAR, and then click the pointer where the new segment should end.



5 Move the Vertex cursor and click to add segments until you have completed the path.

6 Click Done to save the animation and return to the main window.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To test an animation:**



- Click the Run button in the Animation Editor. Alternatively, use the slide bar at the bottom of the Animation Editor to step through an animation sequence.

Creating a cel animation

A cel animation displays a series of images to give the illusion of motion—much like a flipbook animation. For example, you can use cel animation to create the appearance of a spinning globe. The animation consists of different views of the earth, each called a cel. Showing one cel at a time in succession gives the illusion that the globe is turning.



Figure 4
Six cels from a cel animation

After creating all of the individual cels, you arrange them on top of each other in the correct consecutive layer order and then group them. When you play a cel animation, Instructor shows and hides the cells from first to last, over and over.

➤ **To create a cel animation:**

1. In the main window, create the individual objects to serve as cels in the animation.
2. Arrange the layer order of the objects so that the first cel is on the farthest (lowest) layer and the last cel is on the closest (highest) layer.
3. Select all of the objects by holding down the SHIFT key while you click each object with your mouse.
4. From the Object menu, choose Group to group the objects.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

5. Position the objects. For example, in a spinning globe animation, the objects are placed on top of each other so that only the top one is visible. To select and move a single item in the group, double-click it.



6. Click the Properties button on the toolbar, and in the Properties for Group dialog box, assign a name to the group. If you do not assign a unique name to the group, you will not be able to trigger your animation from the Properties for Action Trigger dialog box or from the ToolBook II Actions Editor.

7. Select the group, and then choose Path Animation from the Object menu.

8. *Optional.* Move the grouped objects along a path by creating a path animation as described in the preceding section, "Creating a path animation."



9. Click the Animation Settings button to open the Animation Settings dialog box. Under Cel animation, select On, and then click Close.

10. Click Done in the Animation Editor to save the animation.

➤ **To test an animation:**



- Click the Run button in the Animation Editor. Alternatively, use the slide bar at the bottom of the Animation Editor to step through an animation sequence.

Playing an animation at Reader level

After you create an animation for an object, you can use action buttons to play and reset animations. Use an action trigger to trigger an object's animation and a reset trigger to reset the object's animation.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can also use the Actions Editor to create an action sequence that triggers or resets an object's path or cel animation. One advantage of this method is that you can set the animation to occur as part of a longer sequence of actions, all activated by the same event. You can also create conditional expressions and loops that determine when, and how many times, an animation plays.

Alternatively, you can use OpenScript to play and stop a path animation.

Using an action trigger to play an animation

You set the properties of an action button to define the trigger event and the action that will occur. For example, after you add an action trigger to a page, you can add the object whose animation you want to trigger to the Objects to trigger list in the Properties for Action Trigger dialog box. Then you can define the event that triggers the action and the action that will occur—in this case, animation of the object. To choose the action that will occur for an object, you can click the button under the Objects to trigger list.

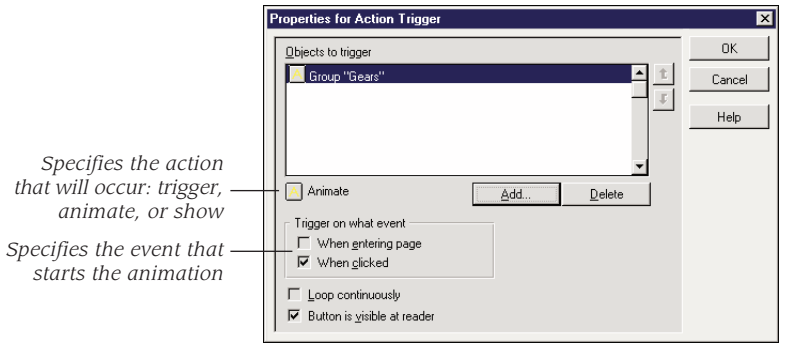


Figure 5
The Properties for Action Trigger dialog box

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can also specify whether you want the action trigger to be visible at Reader level. At Reader level, the animation will either play automatically when entering the page, or interactively when the user clicks the action trigger, depending on what you have specified.

➤ **To play an animation using an action trigger:**

1 Open the Catalog and drag an action trigger onto the same page as the animation you've just created.



2 Select the trigger, and then click the Extended Properties button on the Instructor toolbar.

3 In the Properties for Action Trigger dialog box, under Objects to trigger, click Add. In the Add an Object dialog box, do the following:

   A In the Select object list, select the object for which you created the animation.

   B Under Perform what action, choose Animate.

   C Click OK.

4 In the Properties for Action Trigger dialog box, under Trigger on what event, do the following:

   A Select the When clicked check box, and clear the When entering page check box.

   B Select the Button is visible at reader check box, and clear the Loop continuously check box.

   C Click OK.

5 Switch to Reader level (press F3) to test your animation.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the Actions Editor to play an animation

If you use the Actions Editor to play an object's cel or path animation, you can include the animation as part of a sequence of actions, use a loop to play the animation repeatedly, or specify conditions for playing the animation.

➤ **To create an action sequence that plays an animation:**

1. Select an object (for example, a button or graphic).
2. From the Object menu, choose Actions.

   The Actions Editor appears.
3. In the Event list, select an event that activates the action sequence (for example, On click or On load page).
4. From the Insert menu in the Actions Editor, point to Action, point to Media, and then choose Play Media.

   The Properties for Action dialog box appears.
5. In the Select the media type list, select Animate an object.
6. In the Select an object to animate list, select the object for which an animation has been created; or drag the Choose object button over the object to select it.
7. *Optional.* On the Options tab, specify how many times the animation will play.
8. Click OK.
9. From the File menu in the Actions Editor, choose Update Actions & Close.
10. Switch to Reader level (press F3) to test your animation.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript to play an animation

Path animations can be played and stopped with OpenScript messages. You begin an animation by sending the message to the object whose animation you want to play. Instructor sends the message to move the object along the animation path and to stop the animation when it finishes.

- To play an object's entire path animation, send the *playAnimation* message:

```
send playAnimation 1, this page, true to ellipse "Moon"
```

- To move the object one step along its animation path, send the *playStep* message:

```
send playStep 1 to self
```

Because Instructor sends the *playStep* message to the object for each step of the animation, you can write a handler to modify the animation as it is running. Be sure to forward the *playStep* message if you want the object to continue on its path:

```
--Starts the object's path animation when the object is clicked
to handle buttonClick
   system halfway                --Establishes variable used in playStep handler
   system soundAlreadyPlayed     --Flag indicating if sound clip played
   soundAlreadyPlayed = false
   mmOpen Clip "chimes"
   halfway = (anim_Duration of self) * 1000 / 2
   send playAnimation 1 to self
end
```

(continued)

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

```
--Monitors the animation at each step. When the animation is
--half completed, plays a .wav file.
to handle playStep
   system halfway                --Variable set earlier when animation began
   system soundAlreadyPlayed
   if anim_elapsedTime of target > halfway then
      if soundAlreadyPlayed = false then
         mmPlay clip "chimes"
         soundAlreadyPlayed = true
      end if
   end if
   forward
end playStep
```

- To stop an animation, send the *stopAnimation* message:

```
send stopAnimation to ellipse "Moon"
```

Modifying an animation in the Animation Editor

After you create an animation and play it at Reader level to see how it works, you can modify it using the Animation Editor. For example, you can:

- move an entire path or move one segment of a path using the Move cursor.
- extend a path by adding segments to the end or middle of a path, or shorten a path by deleting segments using the Vertex cursor.
- create, reshape, or delete a curve in a path by modifying segments of it using the Vertex cursor.



For complete step-by-step instructions about creating, modifying, and playing path- or cel-based animations, refer to online Help. To open online Help from the Animation Editor, click the Help button.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Selecting an animation

Before you can modify an animation or its animation settings, you must select the animation you want to edit. There are two ways to select an animation: either in the main window before you open the Animation Editor, or from within the Animation Editor.

➤ **To select an object in the Animation Editor:**

1. With no objects selected in the main window, choose Path Animation from the Object menu.
2. Click the object you want to select in the main window of the Animation Editor.
3. In the Select Object dialog box, do the following:
   - **A** In the Object to animate list, select the name of the object or group to animate.
   - **B** In the Animation list, select the number of the animation to edit.
   - **C** Click OK.



You can click the Select Object button to open the Select Object dialog box, where you can select an object to edit.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Setting animation options

After you create an animation, you can set options that define the behavior and appearance of the animation, such as duration or rate. For example, you can increase the number of steps per second to make an animation appear to run smoother, and you can choose a variable rate to make an animation appear to accelerate or decelerate over its path.

➤ **To set animation options:**

1. Open the Animation Editor and select the animation whose options you want to edit.
2.  Click the Animation Settings button.

   The Animation Settings dialog box appears.
3. Make the changes you want, and then click Close.
4. Click Done in the Animation Editor to save your changes and recompile the animation.

 For more information about animation options, refer to online Help.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 14

Using multimedia

You can play file-based media clips in ToolBook II Instructor—including sound, animation, digital video, and still images. This chapter describes how to use the Universal Media Player object and other Catalog objects to add multimedia flair to your online application.

IN THIS CHAPTER

About multimedia **332**

Using multimedia on the Internet **332**

Choosing a media player **332**

Using the Universal Media Player **335**

Using other media players **343**

Creating and managing clips **347**



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About multimedia

Online applications draw their distinct ability to engage and motivate users from the appeal of multimedia. When onscreen text is joined with audio, video, animation, and graphics, online learning becomes dynamic, effective, and fun. For a discussion of the role multimedia can play in your online courseware, refer to Chapter 3, "Creating multimedia effects and interactive content," in *The click2learn.com Guide to e-Learning*.

Using multimedia on the Internet

You can use media in an application that you plan to deliver via the Internet. If you distribute your application as a native ToolBook II file that users can view with the Neuron browser plug-in, you can expect the media elements you include to perform just as they do in Instructor. If you plan to export your book to DHTML, you will need to use Internet-compatible file formats when choosing media elements. For a discussion of the different types of Internet-compatible media file formats, refer to *The click2learn.com Guide to e-Learning*.

Choosing a media player

The primary device for playing media files (including audio, digital video, and streaming media) is the Universal Media Player: It supports most standard media types, is programmable using the Actions Editor, and retains all functionality when exported to DHTML. Simply drag the Universal Media Player object from the Catalog to your page, specify the media file to be played, and set properties that determine how the media appears and behaves. Alternatively, you can use one of several other media player objects available in the Catalog. These objects offer additional buttons and design options, but more limited functionality.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Choosing a media player

Media Player	Supports	Advantages	Limitations
Universal Media Player	Most currently available digital video, audio, and streaming media formats	Works the same when exported to DHTML as in ToolBook runtime Programmable using the Actions Editor visual programming tool Has a single, consistent authoring interface, which makes it easy to learn and use	Will not display still graphics
Media players	Specific audio, digital video, and still image files, as listed in the Catalog description*	Allow you more control over the appearance of the object on the page in ToolBook runtime	To play certain types of media files, users must load the proper software drivers on their computers Allow limited programming with the Actions Editor visual programming tool

* Will play other file formats if you install proper MCI software drivers

(continued)

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Choosing a media player, *continued*

Media Player	Supports	Advantages	Limitations
Video players	AVI files (digital video)*	Allow you more control over the appearance of the object on the page in ToolBook runtime	To play certain types of media files, users must load the proper software drivers on their computers Allow limited programming with the Actions Editor visual programming tool
Inline clip players	Specific audio, digital video, and still image files, as listed in the Catalog description*	Can be inserted in a text field so that button scrolls with text, similar to a hotword	To play certain types of media files, users must install the proper software drivers on their computers Allow limited programming with the Actions Editor visual programming tool

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the Universal Media Player

The Universal Media Player is the easiest to use and most versatile of all of the media player objects available in the Catalog. It supports a wide variety of digital video, audio, and streaming media file formats. You simply choose the file to display, and ToolBook II automatically presents the appropriate player at Reader level. For example, if you select a Microsoft Streaming Media file, Instructor uses the Windows Media Player ActiveX control to play the file.

Media types supported by the Universal Media Player

If this player is available...	These file formats will play...
RealPlayer	Animated GIF: *.gif Audio files: *.aa Flash files: *.swf Graphic files: *.gif, *.jpeg, *.jpg, *.png MP3 files: *.mp3, *.dat MP3 Playlist files: *.m3u, *.pls RealMedia files: *.ra, *.rm, *.rmj RealMedia metafiles: *.ram, *.rmm, *.rpm RealPix files: *.rp RealText files: *.rt RealText3D files: *.r3t SMIL files: *.smi, *.smil
Flash Player	Flash files: *.swf
Windows Media Player	Audio files: *.aif, *.aifc, *.aiff, *.mid, *.midi, *.rmi, *.wav MP3 files: *.mp3, *.mpa, *.m3u MPEG files: *.mpg, *.mpeg, *.m1v, *.mp2 QuickTime files: *.mov, *.qt Video files: *.avi Windows Media files: *.asf, *.asx, *.wm, *.wma, *.wmv, *.wax

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can find the Universal Media Player in the Media Players category of the Catalog.

➤ **To add the Universal Media Player:**



1. Click the Catalog button on the Instructor toolbar to open the Catalog.
2. In the Media Players category, select the Universal Media Player.
3. Drag the Universal Media Player from the object pane, and then drop it onto your page.

Choosing a media file

After you place the Universal Media Player on a page, you can use the Extended Properties dialog box to choose a media file.

➤ **To choose a media file:**



1. Select the Universal Media Player, and then click the Extended Properties button on the toolbar.



2. On the General tab, click the Browse button.

   The Select a Media File dialog box appears.
3. Navigate to and select the file you want to use, and then click OK.
4. *Optional.* On the Time Markers tab, set time markers to mark positions in the media file with which you want to coordinate other object behavior.

   For more information, see “Using time markers,” later in this section.
5. Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

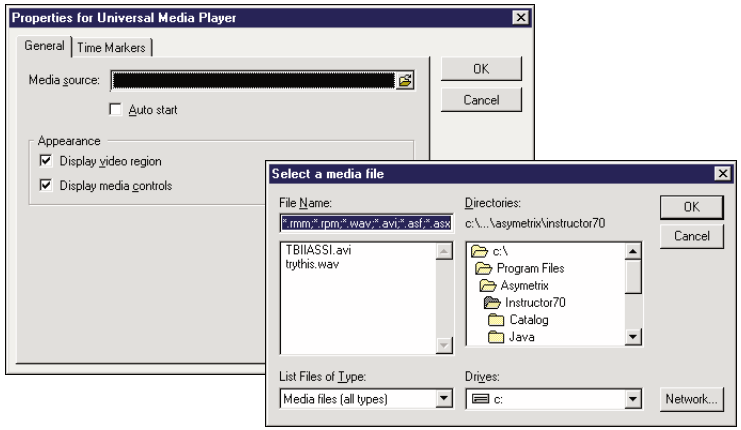


Figure 1
Choosing a file to play in the Universal Media Player

Using time markers

You can use time markers to synchronize the playing of a media file with the behavior of other objects on the page. For example, you can display pop-up text describing a scene at a certain point in a video, and then hide it at a later point. Once you define time markers, you need to create an action sequence in the Actions Editor to handle the media timing event generated each time the file reaches a time marker.

Note If you do not plan to export to DHTML, you can use OpenScript to handle the ASYM_TimeMarker media timing message.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Time markers identify positions in a media file. When a time marker is reached, Instructor sends a media timing event with a parameter named *mediaPosition*. The value of *mediaPosition* is equal to the position in the media file (positions are measured in milliseconds or frames, depending on the media type). There are two types of time markers you can set. You can set time markers that send the media timing event at periodic intervals (for example, every 1000 milliseconds), or you can insert time markers at specific points in the media file.

➤ **To set periodic time markers:**



1. Select the Universal Media Player object, and then click the Extended Properties button on the toolbar.

   The Properties dialog box appears.

2. On the Time Markers tab, do the following:

   A Select Enable time markers.

   B Select Periodic.

   C Type a value for the intervals between the time markers.

3. Click OK.

Most media types measure their positions in milliseconds. Some media types, however, measure their position in frames. The controls on the Time Marker tab of the Properties for Universal Media Player dialog box are labeled according to the position format for the selected media type.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set time markers at specific points in the media file:**



1 Select the Universal Media Player object, and then click the Extended Properties button on the toolbar.

The Properties for Universal Media Player dialog box appears.

2 On the Time Markers tab, do the following:

A Select Enable time markers.

B Select Specific times.

C Do one of the following:



— Click Play, and as the file plays, set a time marker at the current position in the media file by clicking Add time marker.

— Select the field under Marker Time, press CTRL+ENTER (or right-click and select New Time Marker), and then replace Enter Marker with a time in milliseconds or a frame number that corresponds with a position in the media file.

D *Optional.* Enter descriptive text in the Comment field.

E Repeat steps A to C until you have inserted all time markers.

3 Click OK.



Tip To set several time markers at once, play the media file and, while it is playing, click Add time marker as the player reaches the positions of interest in the media file. Once you have set the time markers, you can go back to insert comments or edit the individual time markers.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

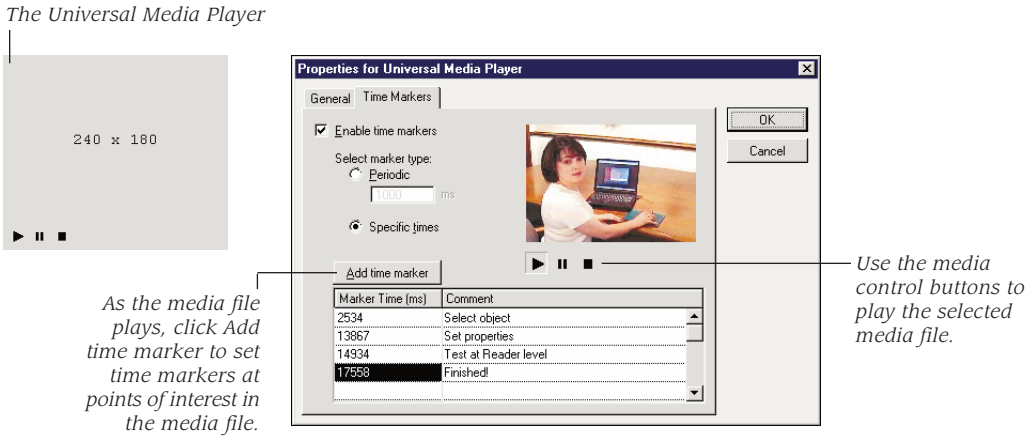


Figure 2
Setting time markers

Creating actions for time markers

For each media file you play, you can set multiple time markers, each of which sends the same media timing event. You can use the Actions Editor to create a single action sequence that runs in response to any media timing event, or you can create an action sequence that responds with different behavior depending on which time marker generated the media timing event.

For more information about the Actions Editor visual programming tool, see Chapter 15, “Using the Actions Editor.”

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To create an action sequence to respond to media timing events:



1. Select the Universal Media Player object, and then click the Actions button on the toolbar.

   The Actions Editor appears.

2. In the Event list, select On media timing event.
3. In the Actions Editor, create an action or series of actions to respond to media timing events.
4. In the Actions Editor, from the File menu, choose Update Actions & Close.

You can also create an action sequence that responds with different behavior depending on which time marker generated the media timing event.

To distinguish one media timing event from another, Instructor sets a parameter named *mediaPosition* to the value (the position in the file) of the time marker that generated the event. For example, if you set a time marker at 1050ms and another at 2050ms, the first time marker sends a media timing event with 1050 as the parameter and the second time marker sends the same event with 2050 as a parameter.

In the Actions Editor, you can distinguish one media timing event from another by creating a conditional expression that compares whether the parameter sent with the event is equal to a specific value (such as 1050). In this way, you can respond to multiple media timing events in one action sequence.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To create an action sequence to differentiate between media timing events:**



1. Select the Universal Media Player object, and then click the Actions button on the toolbar.

   The Actions Editor appears.

2. In the Event list, select On media timing event.

3. From the Insert menu, point to Condition, and then choose If.

   If the Properties for Condition dialog box does not appear, click the Properties button on the Actions Editor toolbar.



4. Click the Build Expression button.

   The Build Expression dialog box appears.

5. Do the following:

   A In the Select the type of item to insert list, select Variables and parameters; in the Select a variable or parameter list, select *mediaPosition*.

   B Insert the cursor at the end of the expression and type = (the equal symbol), and then type in the value of your first time marker.

   C Click OK.

6. Select the conditional expression you just created, and then create an action or series of actions that Instructor runs if that condition is true.

7. Repeat steps 3 through 6 until you have created a sequence of actions to handle each time marker event that you defined.

8. From the Actions Editor File menu, choose Update Actions & Close.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

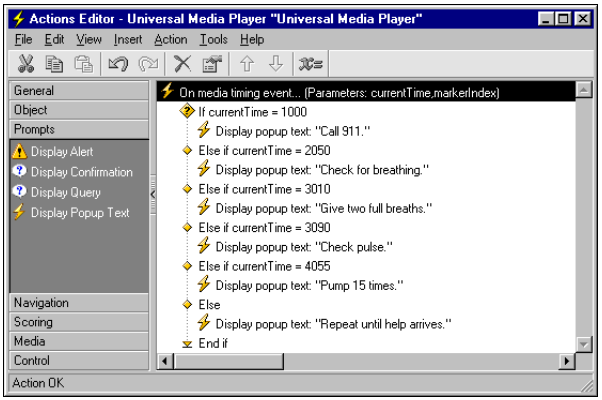


Figure 3
Handling multiple media timing events in an action sequence

Using other media players

The Catalog also includes several other objects that you can use to add multimedia to your application, which function quite differently from the Universal Media Player. These objects play clips, which are named references to a specific segment of a media file, rather than the media file itself. For more information, see “Creating and managing clips,” later in this chapter.

You can set a media player’s properties to refine its appearance (visual media clips only) and behavior. Media players also feature an Extended Properties dialog box: You can select a clip to play, specify when and for how long the clip plays, and (for visual media) determine whether the clip will play in a stage or in a pop-up window.

Note If you do not specify a stage in which to play a visual media clip, it will appear in a pop-up window at Reader level.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with media players and stages

Video players and media players are objects that you drag from the Catalog and drop onto a page in your book. The Catalog also features a stage that you can use in conjunction with a media player to play visual media clips. You can also draw a stage using the stage tool on the tool palette.

➤ **To insert a media player from the Catalog:**



1. Click the Catalog button on the Instructor toolbar to open the Catalog.
2. In the Catalog, select the Media Players category.
3. Drag one of the media player objects from the object pane, and then drop it onto your page.

➤ **To draw a stage:**

Using the interface



1. From the tool palette, choose the stage tool.
2. On your page, hold down the mouse button while you drag, and then release when the stage is the size you want.
3. Using the select tool, click the stage to select it.



4. Click the Properties button on the Instructor toolbar.

   The Properties for Stage dialog box appears.
5. In the Name box, type a name for the stage.
6. *Optional.* Close the Properties for Stage dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- Use the *draw* command with the *stage* object type, followed by the bounds, and then assign the stage a name using the *name* property. For example:

```
draw stage from 1000,1000 to 3000,3000
   name of selection = "videoStage"
```



Note When a video clip is playing in a stage, it is displayed in its own window, which will obscure all objects on the page in that area, even objects that appear in front of the stage when the media is not playing.

Setting the appearance of a stage

To change the appearance of a stage in a video player, you can set properties that define the:

- frame, which is the decorative border around a stage.
- display area, which sets the size and position of visual media clips in a stage.
- transition effects, which determine the effect before and after a visual media clip plays.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Choosing a frame style

You can set a stage frame that draws attention to the stage or that subtly adds dimension to a page.

➤ **To set the style of a stage frame:**



1. Select the stage, and then click the Properties button on the Instructor toolbar.

   The stage may be part of a group—the video player. To select only the stage, double-click the stage.
2. On the Draw tab, under Frame widths, enter the width (in page units) of the outer bevel, border, and inner bevel.
3. *Optional*. Close the Properties for Stage dialog box.

Setting the size and position of clips in a stage

By default, clips have a predefined size and are centered in a stage. If you want to play clips of different sizes in the same stage, you can scale or crop clips so that they fit in the stage. This feature will not export to DHTML.

➤ **To set the size and position of clips in a stage:**



1. Select the stage, and then click the Properties button on the Instructor toolbar.

   The Properties for Stage dialog box appears.
2. Click the Draw tab.
3. In the Sizing list, choose how clips appear in the stage.
4. If you are playing clips of different sizes, select an anchor option from the Anchor list.
5. *Optional*. Close the Properties for Stage dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can also use OpenScript to set the appearance of a stage. For details, refer to the *Programming in OpenScript* online book.

Setting transition effects

To introduce a visual media clip and to smoothly move from one clip to another, you can add a transition effect before and after a clip plays. For example, you can select a transition effect that simulates a page turning or blinds opening.

➤ To set a stage transition effect:



1 Select the stage, and then click the Properties button on the Instructor toolbar.

   The Properties for Stage dialog box appears.

2 Click the PreEffect tab. Then, from the Effect list, select a transition effect to use before the clip plays.

3 Click the PostEffect tab. Then, from the Effect list, select a transition effect to use after the clip plays.

4 *Optional*. Close the Properties for Stage dialog box.

Creating and managing clips

A clip is a reference to any media file—a sound file, video file, animation file, or still-image file. You can use the Clip Manager dialog box to create, select, edit, duplicate, remove, and play the media clips in a book. If you've planned in advance all the audio and visual media you want to include, you can use the Clip Manager to add them to your book all at once.



Note Animations discussed in this chapter refer to animation files—not ToolBook II Instructor animations created with the ToolBook II Animation Editor. For details about ToolBook II Instructor animations, see Chapter 13, "Hiding, showing, and animating objects."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Creating clips

A clip is defined by a file name or media source and start and end points. For example, a clip's start and end points could define the clip as the third song on a music CD or the first 20 frames of a digital video file. A clip could also reference an entire file.

➤ **To create a clip:**

Using the interface

1 Do one of the following to open the Clip Manager dialog box:

- From the Object menu, choose Clips.
- From the Extended Properties dialog box of the media player, click Choose a Clip. In the Choose Media Clip dialog box, click the Clips button.

2 Click New to open the Choose Source Type dialog box, select the type of multimedia you want, and then click OK.

If you are creating a clip from a file, the Choose Source File dialog box appears. Navigate to and select the file you want to use, and then click OK.

3 In the Clip Editor, type a name for the clip you're creating in the Clip box.

4 On the Timing and Options tabs, adjust the settings that apply to your clip, such as volume control, start and end points, and palette options.

5 Click OK to close the Clip Editor.

6 Repeat steps 2 through 5 until you have finished creating clips.

7 Click Close to close the Clip Manager.



Note Start and stop position settings for a clip are not supported in an exported DHTML application. If you export your book to DHTML, the entire media file will play in a Web browser.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

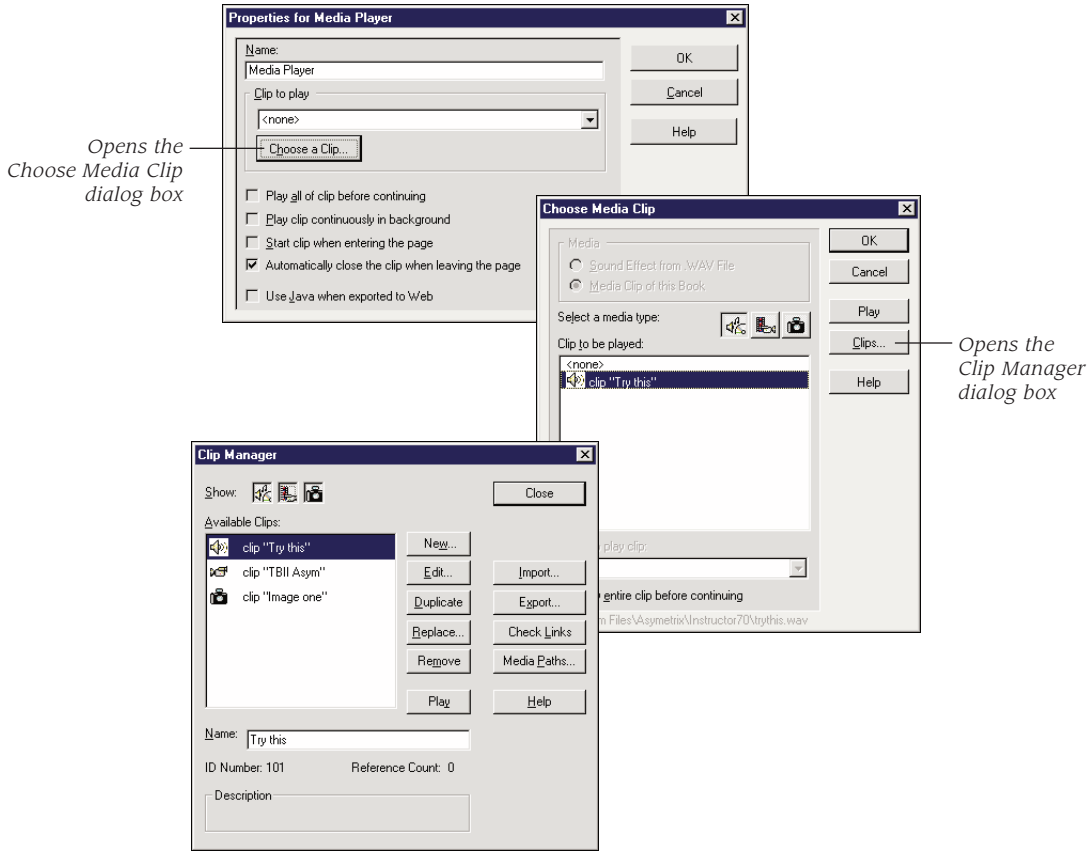


Figure 4
Creating a clip

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

- Create a new clip using the *new clip* command. Set the beginning point using *mmBeginPoint*, the ending point using *mmEndPoint*, and the media search path Instructor will use to locate the media source file using *mmSearchHD* and/or *mmSearchCD*:

```
new clip from "c:\media\horn.avi"
name of It = "tuba"    --Assigns name to clip
mmBeginPoint of clip "tuba" = "00:00:1:15"
mmEndPoint of clip "tuba" = "00:00:2:07"
mmSearchHD of clip "tuba" = true
```

Managing clips

In Instructor, only a reference to a clip is stored in your application; the actual digital data remains stored in the original media source file. When you add a clip to your application, Instructor keeps track of the location of that clip and includes it when you package your book for distribution. If you add a clip to your application and then delete or move the original file, the clip will not play. Therefore, you will want to take care when organizing any media clips you add to your Instructor application.



Tip Remove a clip from your book if you are not using it. Removing unused clips will reduce the size of your application file. Be certain, however, that a clip you remove is not referenced in any script. If you remove a clip that is referenced in a script, you will get an error message when the script attempts to access the clip.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To manage clips:**

Using the interface

1 From the Object menu, choose Clips to open the Clip Manager.

2 In the Available clips list, select the clip you want to work with.

3 Do one of the following:

- To edit the clip, click Edit. The Clip Editor dialog box appears. Edit the clip, and then click OK.
- To make a copy of the selected clip, click Duplicate.
- To replace the selected clip with a clip from a clip library or another book, click Replace. The Replace Clip dialog box appears. Navigate to and select a replacement clip, and then click OK.
- To remove the selected clip from your book, click Remove.
- To play the selected clip, click Play.

4 Click Close to close the Clip Manager.

Using OpenScript

- Remove a clip using the *remove resource* command:

```
remove resource clip "tuba"
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

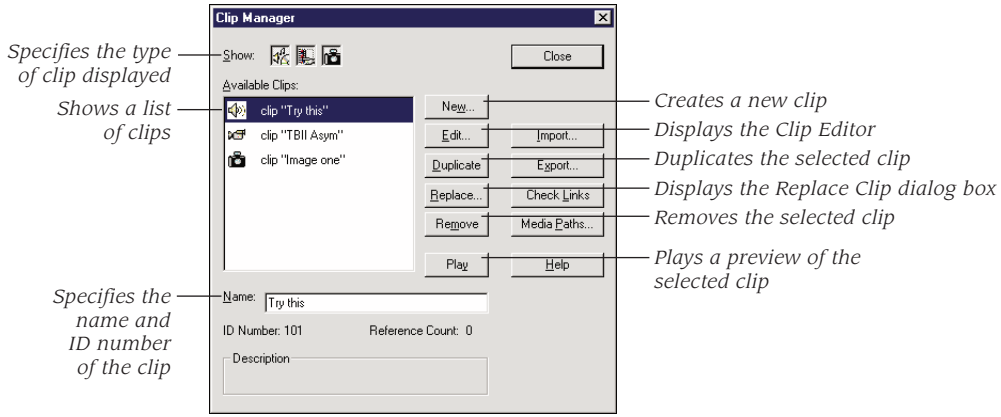


Figure 5
The Clip Manager dialog box

Creating a clip library

A clip library is a file that contains a collection of predefined clips. You can create a clip library to organize frequently used clips. For more information about creating and using a clip library, refer to online Help.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with media clip paths

When you add a media clip to your application, the clip isn't saved in the book; instead, Instructor saves the location of the clip's media source file and, each time the clip is activated, opens and runs that file. The location where the media source file is stored is called its path. If you plan to use the ToolBook II Web Specialist or the ToolBook II AutoPackager to prepare your file for distribution, you don't need to set media paths as you work with clips. The Web Specialist and the AutoPackager will resolve all media path issues for you. However, if you plan to package your application using InstallShield or another setup program, you will need to understand how to work with media clip paths.

Regardless of how you plan to package and distribute your application, you can take one simple step during the development process to ensure that your completed Instructor application can easily locate and play your media files. As you build your application, create an organized directory structure in which media files are stored in a subdirectory of the application file. For example, if your book is stored in C:\Books, store your media files in C:\Books\Media. For more information on creating an organized directory structure, see "Planning your project" in Chapter 2, "Basic concepts."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Making clip paths independent

In order to distribute a multimedia application without knowing in advance where users will install it, Instructor provides media search paths, which are book properties that allow you to store the file name of a media source file separately from its path. The file name is stored with the clip, and you can add the clip's path to a media search path. When you play the clip, Instructor adds the media search path to the clip reference so that it can find the media source file. For example:

- Media search path: C:\Books
- File on the hard disk: C:\Books\Video\Show.avi
- File name stored in clip reference: Video\Show.avi
- Media search path plus clip reference: C:\Books\ + Video\Show.avi

If you create a clip without setting a media search path, the clip reference includes the entire path for the media source file, including the drive letter, subdirectories, and the file name. The media clip will only play on a computer with the same directory structure as the authoring computer.

File organization becomes especially important when you are setting the media search paths for your book. Although we recommend a structure in which your media files are contained in a subdirectory of your book files (for example, Books\Media), Instructor is able to adapt to almost any orderly, consistent organization.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

If you intend to distribute your application on a CD-ROM, replicate the directory and file structure of the CD-ROM on your hard disk and specify the replicated structure on your hard drive as the CD-ROM search path during development. Instructor has two search paths—one for the hard disk and one for the CD-ROM drive—so you can create separate search paths for media that you intend to distribute on CD-ROM. For example, you would list media that will be installed on the user's hard disk in the hard disk media path, while you would list media that will remain on the CD-ROM in the CD-ROM media path. Each search path can store multiple directory names; when Instructor searches the paths, it looks through each directory in turn until it finds the clip.

When you create a new clip, Instructor checks the source file path against the existing media search paths and removes the portion of the clip's path that is already in the search paths. (By default, the current book's path is included in the hard disk search path.) If you store the full path with the clip when you create it but decide later to make the clip path independent, you can change the links to the clip's source file using the ToolBook II AutoPackager. For more information about using the AutoPackager, see Chapter 23, "Preparing applications for release."

When you are ready to distribute your application, you can specify that the media search paths match the directories in which users will install your application. Instructor stores media path information in two book properties and three clip properties.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Media path information locations

Type	Property	Information
Book	*HDMediaPath*	List of subdirectories on hard disk that will be searched to locate a media clip
	CDMediaPath	List of subdirectories on CD-ROM that will be searched to locate a media clip
Clip	*mmSource*	Name of the media source file
	mmSearchHD	Whether to search using *HDMediaPath* (*true* or *false*)
	mmSearchCD	Whether to search using *CDMediaPath* (*true* or *false*)

How Instructor searches for media

When you play a clip, Instructor first checks the clip's *mmSearchHD* property. If it is *true,* Instructor constructs a full path for the clip's media source file by combining values, one by one, from *HDMediaPath* with the clip's *mmSource* property and then attempts to find the clip.

For example, if *HDMediaPath* contains C:\Books, C:\Books\AVI, C:\Books\WAV and the clip's *mmSource* property contains Bells.wav, Instructor searches the following paths in order:

```
C:\Books\Bells.wav
C:\Books\AVI\Bells.wav
C:\Books\WAV\Bells.wav
```

Instructor uses the same search process if *mmSearchCD* is *true* but extracts search paths from *CDMediaPath* instead of from *HDMediaPath.* (If both *mmSearchHD* and *mmSearchCD* are *true*, the paths in *HDMediaPath* are searched first.)

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

There are two very important special values that you can use for HDMediaPath and CDMediaPath: <BookPath> and <CDROM>.

If the media search path contains the string *<BookPath>,* Instructor searches for the path of the current book (the current TBK file). If your media files are in the same directory as the current book or below it, Instructor locates them regardless of where the user installs them. This is the simplest way to handle media paths on a hard drive.

For example, consider the scenario where HDMediaPath contains <BookPath> and <BookPath>\Media and the clip's mmSource property contains Bells.wav. If the book is installed in c:\Books, Instructor first searches c:\Books\Bells.wav, and then it searches C:\Books\Media\Bells.wav. However, if the book is installed in C:\MyFiles\MyBooks, Instructor first searches C:\MyFiles\MyBooks\Bells.wav, and then it searches C:\MyFiles\MyBooks\Media\Bells.wav.

Similarly, if the media search path contains the string *<CDROM>,* Instructor searches all of the CD-ROM drives on the user's computer. If the user has more than one CD-ROM drive, Instructor will search them all. This is the simplest way to handle media paths on a CD-ROM drive.

Both the <BookPath> and <CDROM> special values can be combined with additional path information, as in the following examples:

```
<CDROM>Media
<BookPath>\Media
```

Note the difference in how subdirectories can be added to these special values. The <CDROM> term implicitly includes the backslash directory, so adding a backslash makes the path information invalid. The <BookPath> term does not implicitly include the backslash, so you must add it. For example, the following media search paths are invalid:

```
<CDROM>\Media
<BookPath>Media
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Refining media search paths

A new book's hard disk media search path *(HDMediaPath)* is set to *<BookPath>,* so Instructor automatically finds media source files located in the same directory as the book (the TBK file). You can refine media search paths by adding or removing search paths for both the hard disk and CD-ROM.

➤ **To add or remove media search paths:**

Using the interface

1. From the Object menu, choose Clips to open the Clip Manager.
2. Click the Media Paths button to open the Media Paths dialog box.
3. Under Path to edit, choose either Hard disk or CD-ROM.
4. Do one or more of the following:
   - To add a path, click Add Path, select a new path, and then click OK.
   - To add the current book's path, click Add Book Path. Instructor adds *<BookPath>* to the search path, which is automatically interpreted as the path of the current book (the TBK file).
   - To add the current CD-ROM path, click Add CD-ROM Path. Instructor adds *<CDROM>* to the search path.
   - To remove a path, select a path from the Directories on hard disk media path list, and then click Remove Path.

     If a clip's media source file is in the path you removed, Instructor will not be able to find the file or play the clip.
5. Click OK to close the Media Paths dialog box, and then click Close to close the Clip Manager.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- To add a path to the *HDMediaPath* or *CDMediaPath* book property, use the *push* command:

```
push "c:\media" onto HDMediaPath of this book
```
or
```
push "<BookPath>" onto HDMediaPath of this book
```

- To remove a path from the *HDMediaPath* or *CDMediaPath* book property, use the *clear* command:

```
clear item 2 of HDMediaPath of this book
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

10 Working with buttons and hotwords

11 Working with list boxes and combo boxes

12 Creating a quiz using question objects

13 Hiding, showing, and animating objects

14 Using multimedia

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

PART IV

Adding behavior with action sequences

This part of the ToolBook II Instructor User Guide introduces the Actions Editor visual programming tool. In these chapters, you will learn the basic concepts you need to know to use the Actions Editor. You will also receive step-by-step instructions for creating sample action sequences that add interactive behavior to your Instructor applications.

IN THIS PART

Chapter 15 Using the Actions Editor

Chapter 16 Creating action sequences: Practical examples



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 15

Using the Actions Editor

The Actions Editor is a visual programming tool in ToolBook II Instructor. It features familiar interface elements—such as menus and a toolbar—that you can use to add sophisticated interactive behavior to objects in your Instructor applications. The action sequences that you create in the Actions Editor will export to DHTML automatically when you use the ToolBook II Web Specialist. This chapter describes how to use the Actions Editor to add interactive behavior to your applications.

IN THIS CHAPTER

About action sequences **364**

Choosing when to create action sequences **366**

Opening the Actions Editor **368**

Creating action sequences using the Actions Editor **370**

Creating and using shared action sequences **388**

Managing action sequences **396**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About action sequences

Action sequences add behavior to your ToolBook II Instructor application. Every action sequence consists of one or more behaviors that Instructor carries out in the order you specify. You can create action sequences that prompt users for input, change an object's properties, play media, and much more. You can also repeat actions or specify the conditions under which actions are executed by adding a loop or condition to your action sequence.

In Instructor, you create action sequences using the Actions Editor. The Actions Editor is a tool for visual programming—that is, you specify behavior using an interface that features familiar elements such as menus and a toolbar.

You can create action sequences to program the behavior of any object you add to your Instructor application. An action sequence for an object responds to a specific event. In Figure 1, the event handled by the action sequence for the button is a button click.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

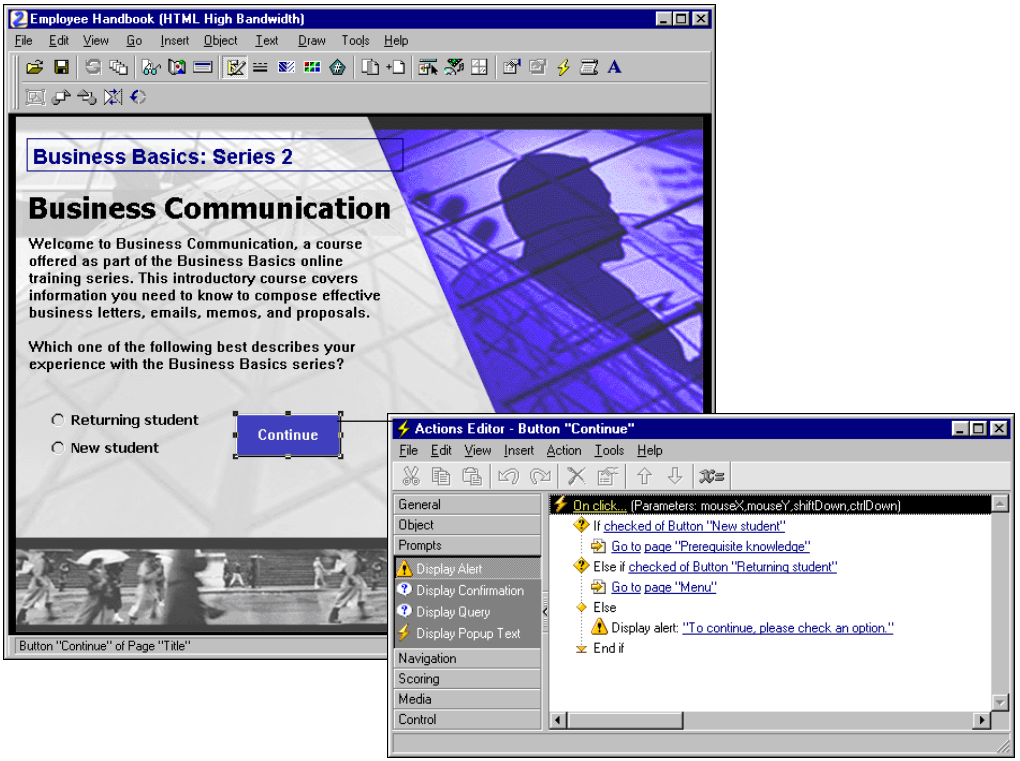


Figure 1
Action sequence for an object

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

You can also create shared action sequences. A shared action sequence is not linked to a specific event and does not belong to any individual object. Instead, you can use a shared action sequence to assign the same behavior to multiple objects. A shared action sequence is stored under a unique name that you assign and is available to be executed as part of any other action sequence. For more information, see "Creating and using shared action sequences," later in this chapter.



Note You can create action sequences to program the behavior of ActiveX controls. For more information about creating action sequences for ActiveX controls, see Chapter 21, "Using ActiveX, Automation, and other Windows technologies."

Choosing when to create action sequences

Action sequences that you build in the Actions Editor work in both native ToolBook II applications and exported DHTML applications. In Instructor, there are other tools you can use to create interactive behavior as well. For example, many Catalog objects have predefined behaviors that may meet your needs. You can also use OpenScript to create and refine behavior for objects if you do not plan to export your book to DHTML.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using action sequences in place of Catalog objects

Many Catalog objects have predefined behaviors. To add simple behavior to your application, a Catalog object is often the best choice. However, for these more complex interactive behaviors, you will find the Actions Editor is a more versatile tool:

- **Creating looping and conditional behavior** Using the Actions Editor, you can execute a series of actions repetitively or in response to specific conditions.
- **Handling certain events** An action sequence can be programmed to handle many events that are not handled by Catalog objects with predefined behaviors. For example, when a mouse is positioned over an object, or when the property of an object changes. You can create an action sequence that handles a generic User event that you send.

Using action sequences in place of OpenScript

Although you can use OpenScript to accomplish the same behavior that you create using action sequences, OpenScript programs do not function on the Web when you export your application to DHTML by using the ToolBook II Web Specialist. By contrast, most action sequences that are exported to DHTML will perform in a Web browser just as they do in the Instructor authoring environment. For details about the performance of specific actions in Web browsers, see the online document named DHTML export considerations.rtf in your Instructor program directory.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Opening the Actions Editor

The Actions Editor is a tool you can use to create an action sequence to handle an event for any object you add to your Instructor application. Once open, the Actions Editor changes dynamically, displaying supported events and any actions you have created for the currently selected object.

➤ **To open the Actions Editor:**

◆ Do one of the following:

— Select the object, and then from the Object menu, choose Actions.



— Select the object, and then click the Actions button on the Instructor toolbar.



— Right-click the object, and then click the Actions button on the toolbar of the right-click menu.

— Select the object, and then press F5.

The Actions Editor appears.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

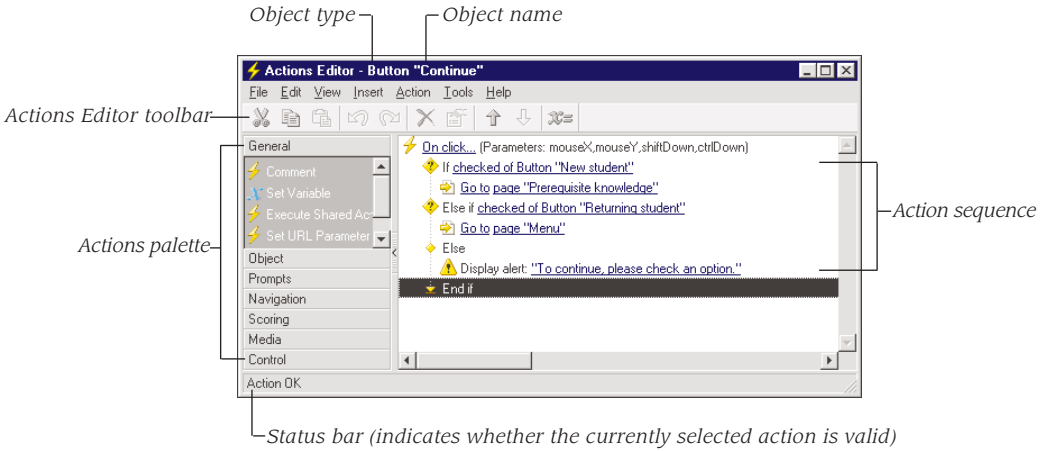


Figure 2
The Actions Editor

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

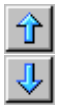
Index

Creating action sequences using the Actions Editor

You create an action sequence in the Actions Editor by adding actions one at a time. After you insert an action into the action sequence, you can move it up or down in the order of actions, as well as cut, copy, or delete the action. You can also select multiple actions at a time to delete, copy, or move them as a group. When running the action sequence, Instructor carries out actions in the order in which they are listed in the Actions Editor window.

In any action sequence, you can modify the items that appear as hotspots in colored, underlined text. When you click one of these hotspots, the appropriate dialog box opens and you can edit the existing settings. You can change the appearance of these hotspots on the Actions tab in the Instructor Options dialog box, which opens when you select Options from the View menu.

Loops and conditions are types of actions with functionality that sets them apart from the other actions. When you insert an action after a condition or loop, Instructor places it in a sublevel (indented) position to indicate that it is carried out only when the loop or conditional expression is true. It also inserts a dashed vertical line connecting the beginning and end of the loop or conditional structure.



You can move an action into or out of a sublevel (indented) position using the Move Up and Move Down buttons on the toolbar.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with the toolbar in the Actions Editor

The buttons on the toolbar are shortcuts to frequently used menu commands in the Actions Editor.

Toolbar buttons in the Actions Editor and their function

Click this button...	To do this...
	Cut the selected action(s) to the Clipboard.
	Copy the selected action(s) to the Clipboard.
	Paste the selected action(s) to the Clipboard.
	Undo the last command
	Redo the last command
	Delete the selected action(s)
	Edit the properties of the selected action(s)
	Move the selected action(s) up one line
	Move the selected action(s) down one line
	Define variables (and parameters for shared action sequences)

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Working with the actions palette

The Actions Editor displays a list of available actions in the actions palette. You can drag an action from the palette and drop it in the action sequence to add a new action.

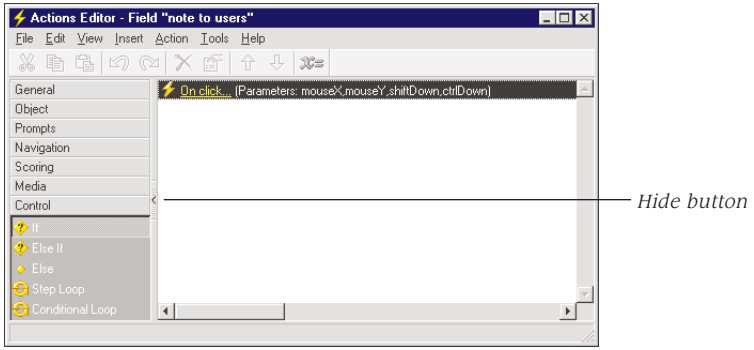


Figure 3
The actions palette

You can resize or close the actions palette. If you close the actions palette, you can select actions from the Insert menu of the Actions Editor.

Choosing an event

Action sequences for objects handle specific events. You can create an action sequence for each event supported by an object. For example, a button might have one action sequence to handle a click and a different action sequence to handle a right-click. You specify the event for which you want to create an action sequence by selecting it from the Event list in the Actions Editor.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

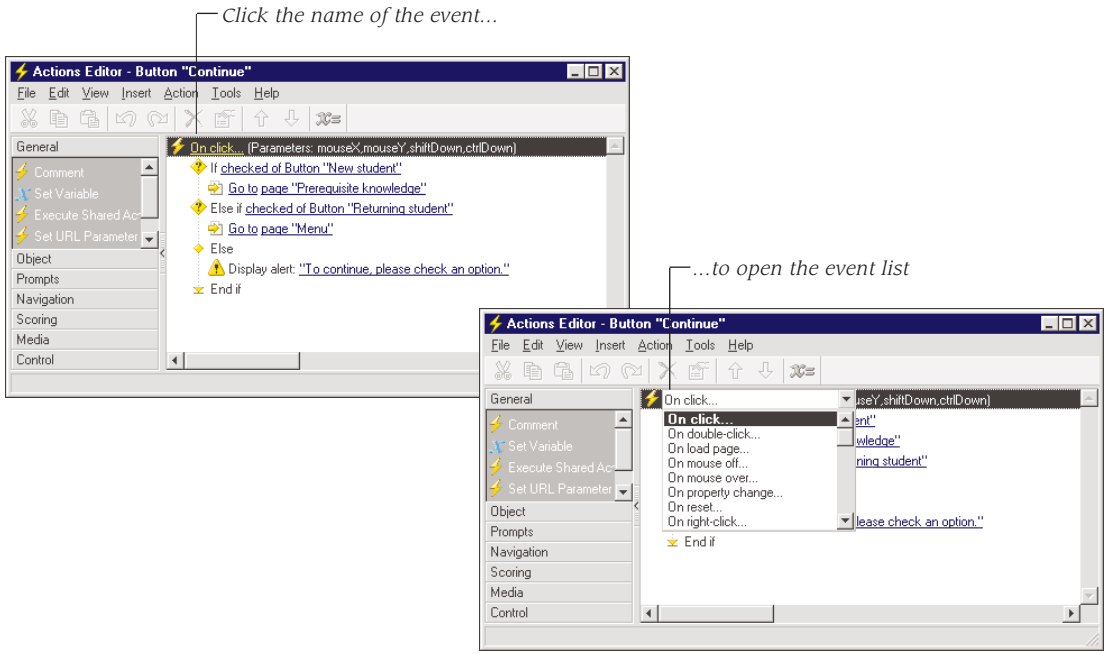


Figure 4
Opening the event list

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To display the Event list for an object:**

1 Select an object in the main ToolBook window.

2 Click the Actions button on the Instructor toolbar to open the Actions Editor.

3 Click the name of the event in the first line of the action sequence.

The Event list opens.

The list of available events depends on the object and includes only events appropriate for the object. For example, since editable fields do not respond to mouse events, the On click and On double-click events are not available.

Common Event list options

Select this Event list item...	To create an action sequence that handles...
On click	A user clicking the object
On User event	A generic event you initiate with the Send User Event action
On double-click	A user quickly clicking the object twice
On load page	A user entering a page
On mouse off	A user moving the pointer off the object
On mouse over	A user positioning the pointer over the object
On property change	A property change
On reset	A page or range of pages being reset
On right-click	A user clicking the right button of the mouse over the object
On trigger	A trigger message
On unload page	A user leaving the page

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Some events (a button click, for example) are initiated by users as they interact with your Instructor application. Other events you create using commands available on the Actions Editor menus; these include Send User Event, Trigger, Set Property, and Reset.

Most events are generated as a result of the user performing some action, or as the result of some change in the system. You can also send a generic event called User event. With the User event, you can send any data in a single parameter. This data may be in the form of a literal value, variable, or expression. The User event is generated by the Send User Event action.



Note For more information about variables and expressions, see "Using variables" and "Building expressions," later in this chapter.

➤ To send the User event in the Actions Editor:

1 From the Insert menu point to Action, point to Object, and choose Send User Event.

The Properties for Action dialog box appears, if this default option is set on the Actions tab of the Instructor Options dialog box.

2 Do the following:

A In the Select the object to receive the User event list, select an object. Or, drag the Choose object button over a named object to select it.

B *Optional.* In the Specify a value to pass as a parameter with this event list, type in a literal value or use the Build Expression dialog box to create an expression to pass as a parameter with the event.

C Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To handle the User event:**

1 Select the object that you specified to receive the User event.



2 Click the Actions button on the Instructor toolbar.

The Actions Editor appears.

3 In the first line of the action sequence, click the name of the event to open the Event list.

4 In the Event list, select On User event.

5 Create an action sequence to respond to the User event.

Inserting actions, conditions, and loops

Action sequences are composed of a variety of actions that are grouped in three functional headings: actions, conditions, and loops. Actions add behavior to your application, conditions specify the circumstances under which actions take place, and loops repeat actions. Every action sequence you create must include at least one action that is not a condition or loop.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To insert an action into an action sequence:



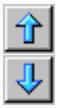
1 *Optional*. If your action sequence will use variables to store values that are used in the action sequence, click the Define Variables button on the Actions Editor toolbar to display the Variables dialog box, define the variables you plan to use, and then click OK.

   For more information about defining variables, see "Using variables," later in this chapter.

2 *Optional*. If you are adding an action to an existing action sequence, select the action after which you want to insert the new action.

3 From the Insert menu, point to Actions, Conditions, or Loops, and then choose the action you want to insert. Or, you can drag one of the actions available in the actions palette to the action sequence.

4 *Optional*. Open the Properties dialog box if it is not already open and select or enter the properties for the action.



5 If necessary, use the Move Up and Move Down buttons on the Actions Editor toolbar to reposition the action in the action sequence.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Actions

Actions add behavior to your application. Some actions are visible to users. For example, you may choose to play media, display a document, or set an object property in response to an event; these all affect how the application appears to the user. Other actions are not immediately visible to the user and are actions that you can use to control how Instructor functions. Setting a variable and executing a shared action sequence are examples of this type of action.

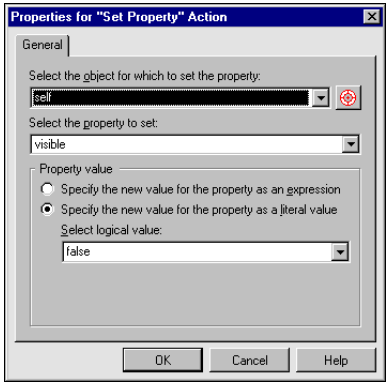


Figure 5
A Properties for Action dialog box

For details about accessing action properties, see “Setting properties for actions” later in this chapter.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Conditions

You can insert a condition into your action sequence to respond to a particular circumstance, such as the state of an object or the selection a user makes. Inserting a condition creates a conditional structure—a junction at which Instructor follows one of two or more distinct paths as it executes the action sequence.

When you insert a conditional If structure, Instructor automatically inserts an End if statement to indicate the end of the structure. Any subsequent conditions you insert in the same structure begin with Else if or, for the final conditional expression, Else. Else is a "catch-all" that covers any condition not specified in an If or Else if statement. Within this conditional structure, Instructor carries out only the sequence of actions directly following the first true conditional expression it encounters. After you insert a condition in an action sequence, you define the condition in the Properties dialog box that appears when you click the *expression* hotword in the action sequence.

Figure 6 shows an action sequence with two conditional structures. The expression of the first structure tests whether the value entered in the text field "Days" is a number from 1 to 7. If this expression is true, Instructor executes the expression of the second conditional structure, which tests whether the value entered in the Text field "Minutes" is a number between 1 and 200. In both structures, if the conditional expression evaluates to false, Instructor prompts the user to enter a value that fits within the defined range.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

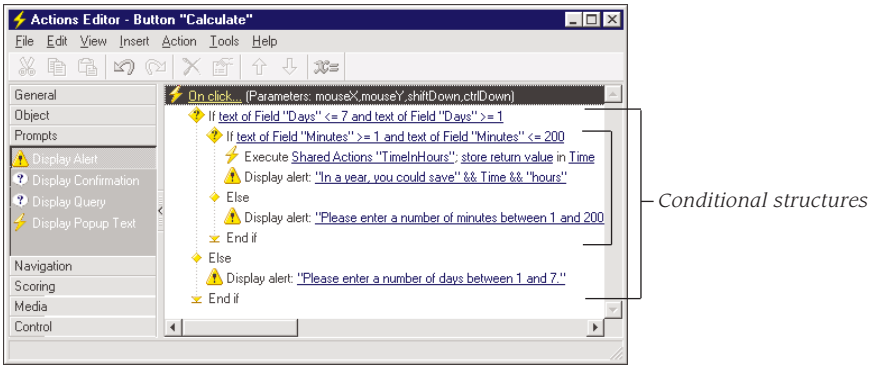


Figure 6
Using conditions in an action sequence

Loops

You can add a loop to your action sequence to run an action or series of actions repetitively. A step loop runs the same action or actions a specified number of times. A conditional loop runs the same action or actions until a condition is met. For example, if you want to keep prompting users for input until they provide the correct answer, you can use a conditional loop.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

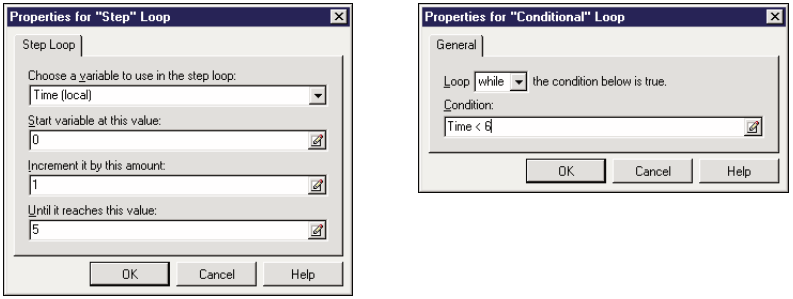


Figure 7
The Properties for Step Loop and Properties for Conditional Loop dialog boxes

Setting properties for actions

For each action, there is a Properties dialog box where you can set options and refine behavior. The options displayed in this dialog box vary depending on the behavior you have chosen to insert.

➤ **To open the Properties dialog box for an action:**

◆ In the Actions Editor, do one of the following:



— Select the action, and then click the Action Properties button on the Actions Editor toolbar.

— Click the name of the event on the first line of an action sequence.

— Select the action, and then choose Properties from the Action menu.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

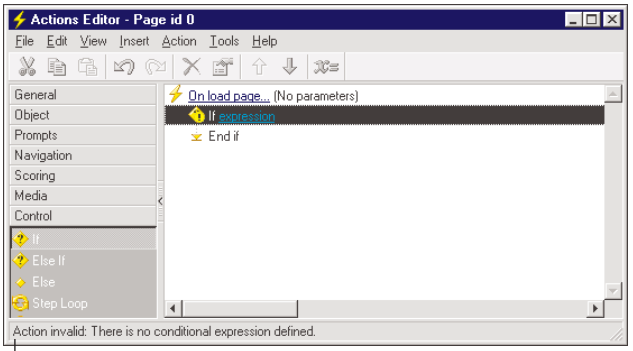
PART VI
Distributing and printing your application

Glossary

Index

Creating valid actions

An action will function at Reader level only if it is valid—that is, if you have provided all the information necessary for the action to run. Instructor provides immediate feedback if the actions you create are invalid. As you create an action sequence, Instructor displays a small warning symbol beside an invalid action and describes why the action is invalid in the status bar.



Status bar displays an explanation

Figure 8
Invalid action

You can also check the validity of all actions in the book at once using the Validate Actions command: on the Tools menu point to Actions, and then choose Validate Actions. For more information about valid actions, see “Validating actions,” later in this chapter.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Building expressions

When you create an action, you can use an expression in place of a literal value such as a number or word. In an expression, you can combine values (variables, object properties, and literal values) using operators and functions to produce a result.

There are three broad types of expressions:

- **arithmetic expressions** These are expressions that use arithmetic operators and functions to combine numbers (or variables and object properties whose values are numbers) and produce a numerical result. The following are examples of arithmetic expressions:

```
5
b^2-4*a*c
```

- **logical expressions** (also called conditional expressions) These are expressions that typically compare values and return a true or false value. The values that are compared can be expressions (arithmetic, logical, or text). Conditions and conditional loops require logical expressions. The following are examples of logical expressions:

```
enabled of button "advanced"
(Variable1 + Variable2) <10
"oscilloscope" is in studentResponse
```

- **text expressions** These are expressions that work on and return text values. For example, you can join two text values using the & (ampersand) operator or extract a single character or string of characters from a block of text using the character and characters operators. The following are examples of text expressions:

```
characters 1 to 5 of userEntry
charCount(userEntry)
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

When you are creating an action, you can type an expression anywhere a value is required. Clicking the Build Expression button displays the Build Expression dialog box, which helps you create a valid expression. The elements from which you can build an expression are grouped as follows: Operators and functions, variables and parameters, object properties, and special values.

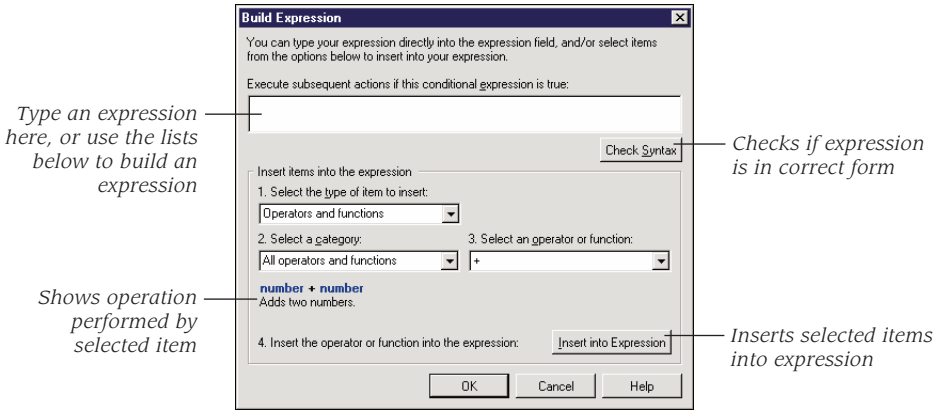


Figure 9
The Build Expression dialog box

Operators and functions

An operator is a symbol or word that causes Instructor to perform an action on operands, resulting in another value. For example, in the expression Variable1 < Variable2, the operator is the < symbol, and Variable1 and Variable2 are operands. A function is an operation that returns a value. For example, the function round() returns the number (or results of an operation) supplied in parentheses rounded to the nearest integer.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Variables and parameters

A variable is a named container that can hold any value. You can use the Set variable action to assign a value to a variable in an action sequence. For more information, see "Using variables," later in this chapter.

A parameter is a read-only value defined for a specific event or shared action sequence. Certain events have parameters that supply additional information about the event. For example, the parameter ctrlDown defines whether the CTRL key was pressed when an event (such as a button click) occurred. The parameters available for a given event are listed in parentheses after the event name in the Actions Editor. When you create a shared action sequence, you can define parameters that the shared action sequence uses to complete any operations it performs. For more information, see "Creating and using shared action sequences" later in this chapter.

Object properties

You can use an object property in an expression anywhere you can use a literal value or variable. In the Build Expression dialog box, you can select the page and background in which you are working, all named objects on the page, self, and target. Target refers to the object that first received the message that generated the event. Self refers to the object to which the action sequence belongs. The options available in the Property list change depending on the object you select.

Special values

Special values are variables with system-defined values that are available in different contexts. For example, the special value pi represents the trigonometric constant pi (approximately 3.14159).

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using variables

Variables are named containers that store values that can be modified while an action sequence runs. In the Actions Editor, you can define two types of variables: local and global. You can use a local variable only as part of the action sequence in which you defined it; so, local variables in different action sequences are unrelated even if you give them the same name. A local variable is reset to its initial value each time the action sequence runs.

Global variables may be used by any action sequence. If an action sequence contains a local variable with the same name as a global variable, the local variable is used when the action sequence runs.

➤ **To define a variable:**



1 In the Actions Editor, click the Define Variables button on the toolbar.

   The Variables dialog box appears

2 On either the Local Variables or Global Variables tab (depending on the type you want to define), click New.

3 In the Variables dialog box, type the name of the variable, and then click OK.

4 In the Initial Value box, type a value.

5 Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

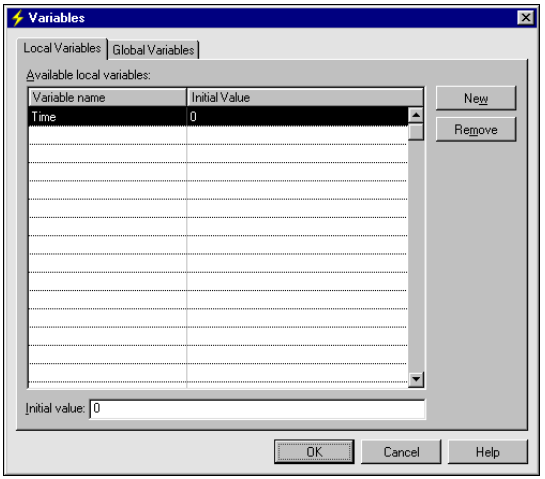


Figure 10
The Variables dialog box

Using arrays

You can define an array to store related data in one variable. An array has a variable name followed by an index in brackets. The index specifies which element of the array is being referenced. For example, in an array named city, city[3] references the value stored in the third element of the array, which could store a value such as “Seattle”.

In the Actions Editor, an array can have only a single dimension. The array index can be numeric or any string that does not contain a space, comma, or tab. In the following examples the array index is a string of characters:

PersonalInfo[“FirstName”]
PersonalInfo[“LastName”]
PersonalInfo[“Age”]

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To define an array:**

1. In the Actions Editor, click the Define Variables button on the toolbar.

   The Variables dialog box appears.

2. On either the Local Variables of Global Variables tab (depending on the type you want to define), click the New button.

3. Enter a name for the array.

4. Select the checkbox labeled Make this variable an array, and then click OK.

You can refer to an element of an array in an expression or in any action that allows you to set a variable. In actions that allow a variable to be set, such as Set Variable or Score Pages, you can choose array variables from a list or you can type in an array name followed by the specific element in brackets.

Creating and using shared action sequences

In most ways, creating a shared action sequence is identical to creating an action sequence for an object. Refer to the sections above to learn about using variables, building expressions, and inserting actions into your shared action sequence.

This section addresses the differences between the two types of action sequences and what you need to know to use shared action sequences in your application.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Naming and locating shared action sequences

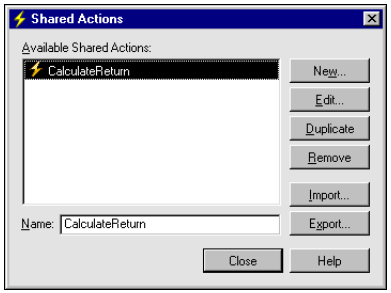
Shared action sequences are stored together under unique names. You work with shared action sequences in the Shared Actions dialog box. You can access the Shared Actions dialog box from the main Object menu or the Actions menu in the Actions Editor.

➤ **To create a new shared action sequence:**

1 Do one of the following:

- From the Object menu, choose Shared Actions.
- In the Actions Editor, from the Action menu, choose Shared Actions.

The Shared Actions dialog box appears.



2 Click New.

The Actions Editor appears.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using shared action sequences

You can use a shared action sequence to add the same behavior to multiple objects. Shared action sequences are comparable to subroutines or functions in most programming languages and are very powerful and useful. Instead of creating the same sequence of actions multiple times, you create a single shared action sequence and call it as many times as you need it. Using shared action sequences also facilitates editing and troubleshooting. If you discover a problem with a shared action sequence, or need to alter its behavior, you only need to make one change rather than many.

You can also use a shared action sequence (like a function) to perform an operation and return a value to the action sequence that calls it. When you create a shared action sequence for this purpose, you can define parameters that the shared action sequence uses to complete the operation.

For example, in Figure 11, a shared action sequence is used to calculate the commuting hours a driver could save in a year by telecommuting (working from home) one day per week. The shared action sequence, TimeInHours, which performs the calculation, uses two parameters: Days and Minutes. To divide the calculation into manageable blocks, it uses two local variables, called AvgWeek and ShortWeek, to store values while the action sequence runs. After it completes the operation, the shared action sequence returns the result (the number of hours saved in a year) to the action sequence that called it. It uses a variable called returnValue to store the value that will be returned.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

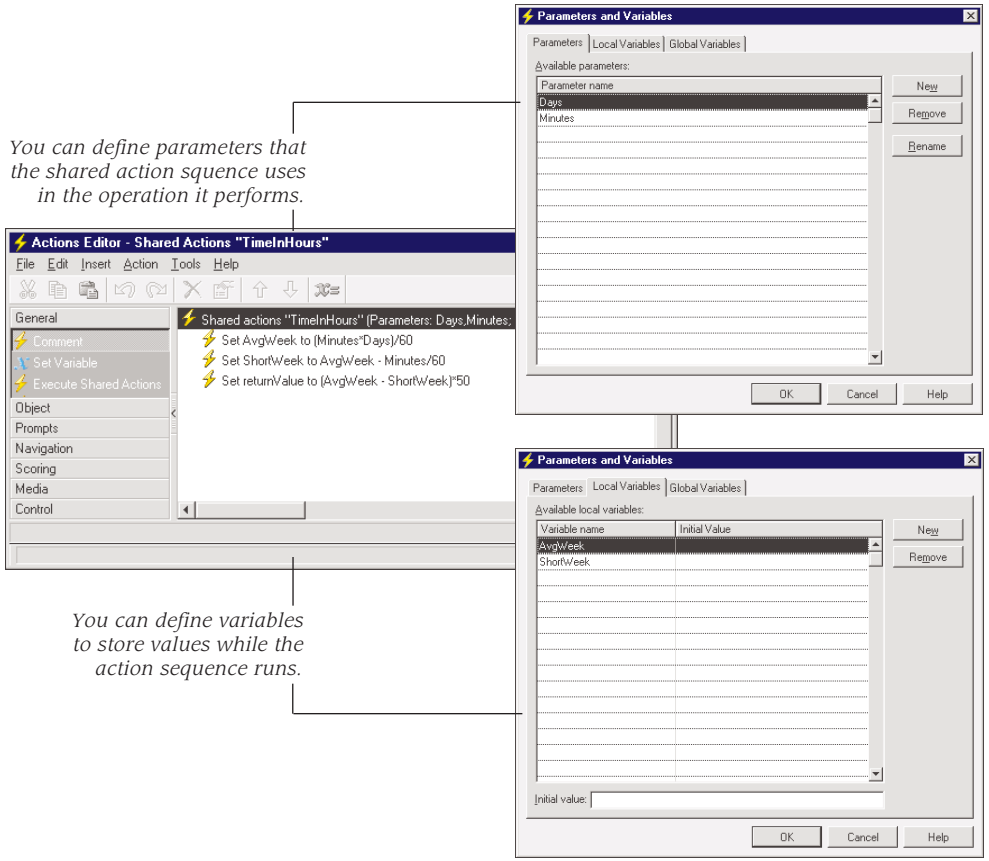


Figure 11
Creating a shared action sequence

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To define parameters for a shared action sequence:**

1 Open or create a new shared action sequence.



2 Click the Define Variables and Parameters button on the Actions Editor toolbar.

The Parameters and Variables dialog box appears.

3 On the Parameters tab, click New.

4 In the Parameters dialog box, enter a descriptive name, and then click OK.

5 Repeat steps 3 and 4 until you have defined all the parameters your shared action sequence needs to complete any operations it performs.

➤ **To define a variable for a shared action sequence:**

1 Open or create a new shared action sequence.



2 Click the Define Variables and Parameters button on the Actions Editor toolbar.

The Parameters and Variables dialog box appears.

3 On the either the Local Variables or Global Variables tab (depending on the type of variable you want to define), click New.

4 In the Variables dialog box, enter a descriptive name, and then click OK.

5 *Optional.* Type a starting value for the variable in the Initial value box.

6 Repeat steps 3, 4, and 5 until you have defined all the variables used in your shared action sequence.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set the return value:**

1. Create a new shared action sequence.
2. In the Actions Editor, from the Insert menu point to Action, point to General, and then select Set Return Value.

   The Properties for Action dialog box appears.
3. Type in the value to return to the action sequence or use the Build Expression dialog box to build an expression, and then click OK.

Executing shared action sequences

A shared action sequence executes only when an action sequence or another shared action sequence calls it.



Note If you execute a shared action sequence that returns a value, you should define a variable to store that value.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To execute a shared action sequence:**

1 Open the action sequence in which you want to execute a shared action sequence.

2 Select the action after which you want the shared action sequence to run.

3 From the Insert menu point to Action, point to General, and then choose Execute Shared Actions.

The Properties for Action dialog box appears.

4 Do the following:

A In the Select a shared action sequence list, select the shared action sequence.

B *Optional*. If the shared action sequence uses parameters, enter a value (as a literal value or expression) for each parameter in the Parameter Value field. For more information, see "Using parameters," later in this section.

C *Optional*. In the Specify a variable to store the return value list, select a variable name.

5 Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

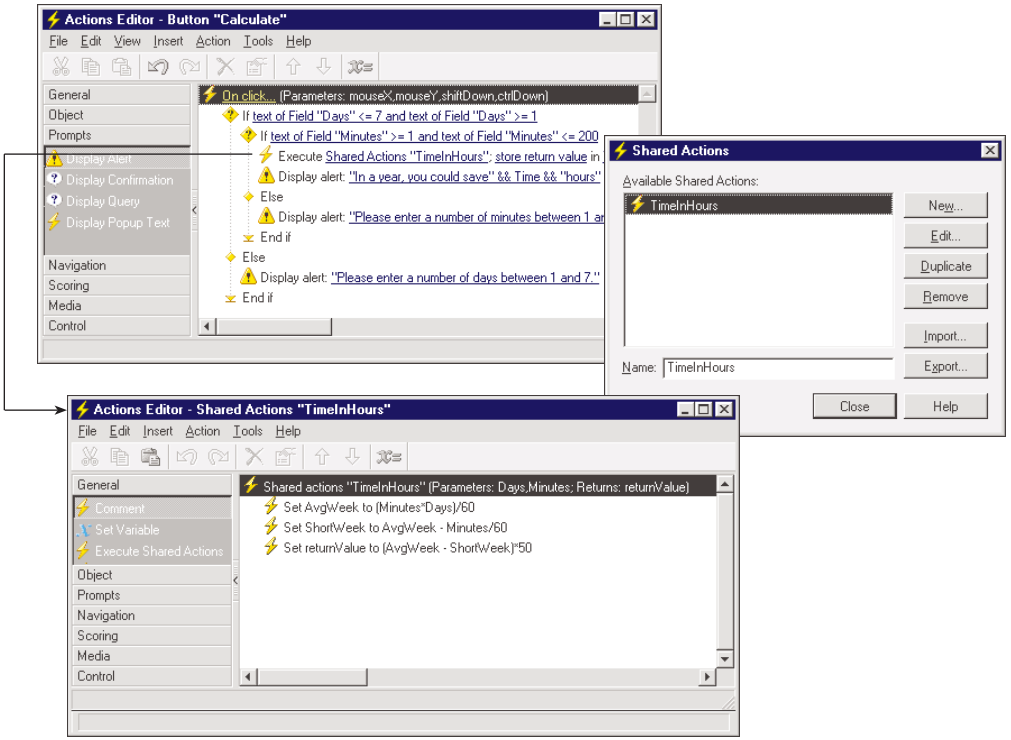


Figure 12
Executing a shared action sequence

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using parameters

When you create a shared action sequence, you can define parameters that the shared action sequence uses to complete any operations it performs. When you call the shared action sequence from another action sequence, you can supply values for those parameters. These may be literal values, variables, object properties, or expressions. The parameter list is only enabled if the shared action sequence accepts parameters.

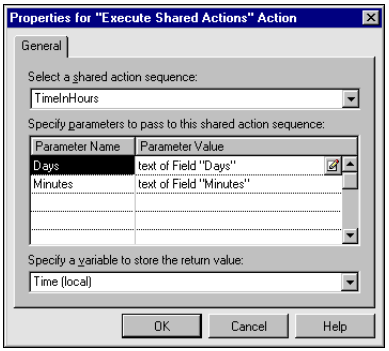


Figure 13
Sending parameters to a shared action sequence

Managing action sequences

Instructor provides tools you can use to validate actions, to view action sequences in a text file, and to test your action sequences as you build your application. You can also import or export an action sequence to reuse it in another book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Validating actions

Only valid actions will run at Reader level. When you finish authoring, you can check the validity of all the actions you've created for a group of objects, a page, a background, or the entire book.

➤ **To validate actions:**

1. From the Tools menu, point to Actions, and then choose Validate Actions. (You can also access this command from the Tools menu in the Actions Editor.)
2. In the Validate actions for list, select the object(s) to check.
3. *Optional.* Check Validate child objects .
4. Click Validate Now.

   The Validate Actions dialog box expands to display a list of invalid actions, the objects and events they handle, and a description of the problem that invalidates the action.
5. To edit an action, double-click it in the list.

   The Actions Editor appears, and the invalid action is highlighted.
6. Correct the action, and then click Close.

An action can be technically valid, but not produce the behavior you expect. Before you distribute your application, test all action sequences at Reader level (and/or in a Web browser) to ensure that the actions perform as you want them to.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Viewing action sequences

Large, complex applications may contain dozens of different action sequences, all of them adding different behaviors to your application. Fortunately, you don't have to open dozens of Actions Editor windows to find a particular action sequence. You can use the Dump Book Actions command to create a text file containing all the action sequences in your book, or you can view all the action sequences on a page using the View Page Actions command.

➤ **To save the action sequences in your book in a text file:**

1. From the Tools menu, point to Actions, and then choose Dump Book Actions. (You can also access this command from the Tools menu in the Actions Editor.)
2. In the Dump Book Actions dialog box, specify a name and location for the text file.
3. Click OK.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

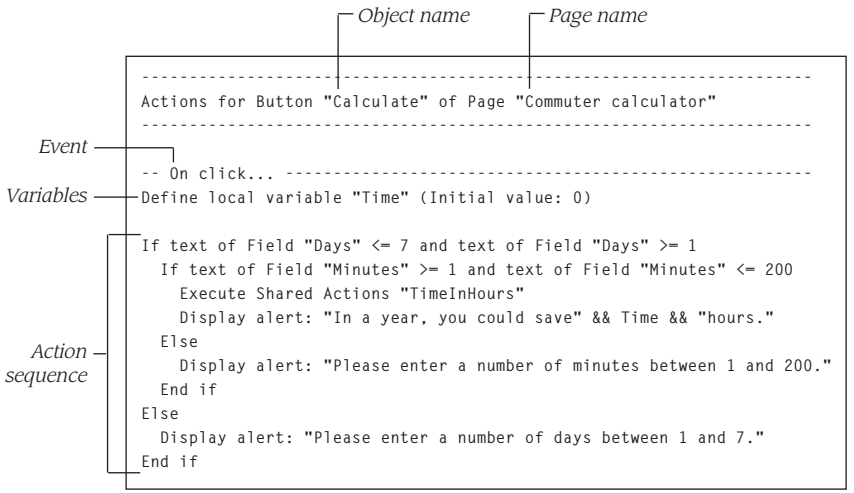


Figure 14
Actions displayed in a text file

➤ **To view all the action sequences on a page:**

1 From the Tools menu, point to Actions, and then choose Show Page Actions.

The Show Page Actions dialog box appears.

2 Do one of the following:

- Click Print to send the document to your default printer.
- Click Copy to copy the contents of the window to the Clipboard. You can then paste the actions into any text document.

3 Click Close.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Enabling and disabling actions

You can disable any action in an action sequence. You can disable actions to isolate others for testing or, if you choose to remove some behavior from an action sequence, you can disable an action instead of deleting it (this is equivalent to commenting out a line of code).

➤ **To disable or enable actions in an action sequence:**

1. Select one or more actions in the action sequence. To select multiple actions, press CTRL while you click the actions.
2. In the Actions Editor, from the Action menu, choose Enabled.

   A check mark appears beside the command when the action is enabled. Disabled actions appear in gray in the Actions Editor.

Importing and exporting action sequences

You may want to use the same action sequence in more than one Instructor book. You can use the same sequence of actions in more than one book by using the Import and Export commands. When you export a sequence of actions, Instructor saves the exported actions in a file with a .tba extension in the location you specify.



Note If the action sequence you are importing contains explicit object references or uses variables, be sure that the book or action sequence into which you are importing the file contains the same object names and variable names.

In the Actions Editor, you can select and export specific actions. You can also import actions and insert them after a selected action in the action sequence. In the Shared Actions dialog box, you can export an entire shared action sequence or create a new shared action sequence by importing actions.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Importing and exporting actions in the Actions Editor

When you export actions from the Actions Editor, Instructor saves only those actions you specifically select in a file with a .tba extension. When you import a .tba file into the Actions Editor, Instructor copies all the actions into the file after the currently selected action in the action sequence.

➤ **To export actions in the Actions Editor:**

1. Open the action sequence from which you want to export actions.
2. Select the actions you want to export. To select multiple actions, press CTRL while you click the actions.
3. From the Actions Editor File menu, choose Export Actions.

   The Export Actions dialog box appears.
4. Navigate to the file location where you want to save the .tba file, type a name for the file, and then click OK.

➤ **To import actions in the Actions Editor:**

1. Open the action sequence into which you want to import actions, or open the Actions Editor for a new action sequence.
2. If importing into an existing action sequence, select the action after which you want to insert the actions stored in the .tba file.
3. From the Actions Editor File menu, choose Import Actions.

   The Import Actions dialog box appears.
4. Navigate to and select the .tba file you want to import, and then click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Importing and exporting from the Shared Actions dialog box

You can export an entire shared action sequence at once using the Export command in the Shared Actions dialog box. You can also import any .tba file as a new shared action sequence using the Import command.

➤ **To export a shared action sequence in the Shared Actions dialog box:**

1. From the Objects menu, choose Shared Actions.

   The Shared Actions dialog box appears.

2. Select the shared action sequence you want to export, and then click Export.

   The Export Shared Actions dialog box appears.

3. Navigate to the file location where you want to save the .tba file, type a name for the file, and then click OK.

➤ **To import in the Shared Actions dialog box:**

1. From the Objects menu, choose Shared Actions.

   The Shared Actions dialog box appears.

2. Click Import.

3. Select the .tba file you want to import, and then click OK.

4. In the Shared Actions dialog box, in the Name box, select NewSharedAction, and replace it with a descriptive name.

5. *Optional.* Click Edit to make any changes to the new shared action sequence you imported.

6. Click Close.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Chapter 16

Creating action sequences: Practical examples

There are thousands of different ways you can combine actions in action sequences to add behavior to your Instructor application. This chapter takes you through the steps involved in creating action sequences that handle some common scenarios.

IN THIS CHAPTER

About the examples **404**

Using the examples **405**

Example 1: Using conditions **407**

Example 2: Using variables **414**

Example 3: Changing object properties **421**

Example 4: Using shared action sequences **426**

Troubleshooting action sequences **441**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

About the examples

The examples presented in this chapter provide a hands-on introduction to creating action sequences using the Actions Editor. As you work with the Actions Editor, you will learn to combine actions in many ways to create powerful results. Each of the examples that follow focuses on learning a particular skill—using variables for example. All four of the examples demonstrate one or more of the following:

- Defining and using variables and parameters
- Adding actions that display text, change object properties, and send users to certain pages
- Handling multiple events
- Testing conditions
- Building expressions
- Using shared action sequences

For more information about using the Actions Editor to create action sequences, see Chapter 15, “Using the Actions Editor.”



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Using the examples

You can easily recreate the examples in this chapter because most of the work of building the pages has been done for you in the Actions Editor Walkthrough book. All you have to do is open the book and create the action sequences that add behavior to the objects.

This chapter assumes you are familiar with basic Instructor concepts. Before you begin, you should feel comfortable switching between Author and Reader level, adding pages and backgrounds, adding objects to pages using the Catalog or the tool palette, and using the Properties dialog box to refine an object's appearance and behavior.

Since the examples that follow are self-contained, you can complete them in any order you like. Each example takes between five and twenty minutes to complete, depending on your level of experience with Instructor and your knowledge of basic programming conventions. When you finish an example, you can view the results immediately.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

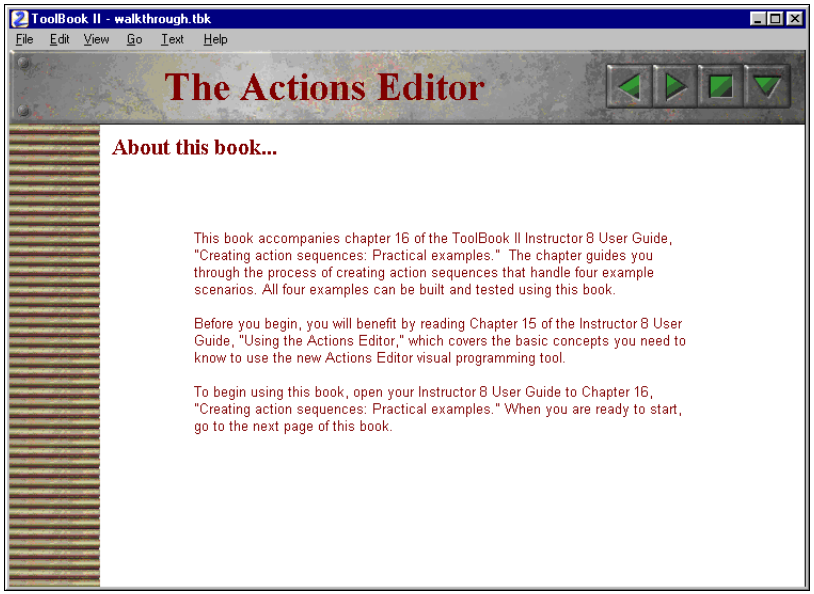
Index

➤ **To open the Actions Editor Walkthrough book:**

1 Start Instructor.

2 On the Welcome tab of the ToolBook II Instructor Startup dialog box, select Walkthrough.tbk, and then click OK.

Instructor opens the book shown below.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Example 1: Using conditions

There are situations in which you will want different types of users to view, or have access to, different information. This example uses conditions in an action sequence to send new users to a page containing introductory text while sending advanced users directly to a page with a table of contents.

Open the second page of the Actions Editor Walkthrough book. Note that there are three interactive objects: two radio buttons (labeled New user and Advanced user) and a pushbutton (labeled Continue). You will create an action sequence for the Continue button that sends users to different pages of the book depending on their radio button selection.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Note To work with an object in the Actions Editor, you must assign it a unique name.

Figure 1 shows the action sequence applied to the Continue button for Example 1. Instructor executes the action sequence as follows: First, Instructor evaluates which one of two radio buttons a user selected. Depending on the radio button selection, Instructor sends the user to a specific page. If the user did not select a radio button, Instructor prompts the user to check a button and try again.

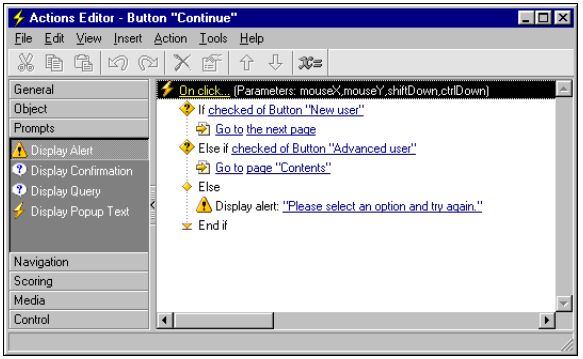


Figure 1
Full action sequence for Example 1

Tip As you work through the following procedures, you can refer back to Figure 1 for an example of how the action sequence should look.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Building the action sequence

Building this action sequence shows you how to:

- select an event.
- define conditions for behavior.
- go to pages in a book.
- display a message in a dialog box.

Opening the Actions Editor and selecting an event

Every action sequence you create for an object handles a specific event and you can create a separate sequence of actions for each possible event. In this case, you will create an action sequence that handles a button click.

➤ **To open the Actions Editor and select the event:**

1. If Instructor is currently open at Reader level, press F3 to move to Author level.
2. On the second page of the Actions Editor Walkthrough book, select the Continue button.
3. From the Object menu, choose Actions.

   The Actions Editor appears.
4. If On click is not selected in the action sequence, click the event to open the Event list, and then select On click.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Inserting actions

Because you want to send users to two different locations depending on their radio button selections, you need to determine which button a user has selected. You can insert a condition to create a conditional expression that tests user selection.

➤ **To insert a condition:**

1. From the Insert menu point to Condition, and then select If.
2. If the Properties for Condition dialog box does not open automatically, click the Properties button on the Actions Editor toolbar.



3. Click the Build expression button at the end of the Condition box to open the Build Expression dialog box.
4. In the Build Expression dialog box, do the following:
   - **A** In the Select the type of item to insert list, select Object properties.
   - **B** In the Select an object list, select Button "New user." Or, drag the Choose object button over the New user radio button to select it.
   - **C** In the Select a property list, select checked.
   - **D** Click Insert into Expression.
5. *Optional.* Click Check Syntax to check the syntax of your expression.
6. Click OK to close the Build Expression dialog box, and then click OK again to close the Properties dialog box.

   Instructor inserts the expression you created and an End if statement to indicate the end of the conditional structure.
7. Using the skills learned above, create a second conditional expression directly below the first: Else if checked of Button "Advanced user."
8. From the Insert menu, point to Condition, and choose Else (note: since Else always covers all other possible conditions, the Properties for Condition dialog box will not appear).

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Note The Else condition covers any other possible condition. In this example, the only possible other condition is that neither radio button is checked.

Once you have determined which radio button the user selected, you need to send the user to the appropriate page of the book. You can use the Go to page action to send a user to a particular page.

➤ **To insert the Go to page action:**

1 Select the first conditional expression in the action sequence (If checked of Button "New user").

2 From the Insert menu, point to Action, point to Navigation, and choose Go to Page.

3 In the Properties for Action dialog box, do the following:

   A In the Select the type of navigation list, select Jump to page.

   B In the Select the page to navigate to list, select Next page.

   C Click OK.

4 Select the second conditional expression (Else if checked of Button "Advanced user"). From the Insert menu, point to Action, point to Navigation, and then choose Go to Page.

5 In the Properties for Action dialog box, do the following:

   A In the Select the type of navigation list, select Jump to page.

   B In the Select the page to navigate to list, select Specific page.



   C In the Select a specific page box, click the Choose page button.

   D In the Choose Page dialog box, select Page "Contents," and then click OK.

6 In the Properties for Action dialog box, click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

If a user fails to select a radio button before clicking continue, you need to prompt the user to select an option. There are several options you can choose from to display a message to users. This action sequence uses the Display alert action to create a dialog box with a note to the user.

➤ **To insert the Display alert action:**

1 Select the Else condition.

2 From the Insert menu, point to Action, point to Prompts, and then choose Display Alert.

3 In the Properties for Action dialog box, do the following:

   A Select Enter literal text.

   B In the Specify the text to display box, type *Please select an option and try again.*

   C Click OK to close the Properties dialog box.

4 In the Actions Editor, from the File menu, choose Update Actions & Close.



Tip You can leave the Actions Editor open while you test an action sequence. From the File menu, choose Update Actions rather than Update Actions & Close.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Testing the action sequence

It's a good idea to test your action sequences as you build them. You can ensure that an action sequence works and that you are satisfied with the results before you use it again or develop other behaviors to respond to the expected outcome.

➤ **To test the Display alert action:**

1 Press F3 to switch to Reader level.

2 Click Continue.

If a dialog box appears with the message, "Please select an option and try again," it indicates that both your conditional structure and Display alert action are working. Now test the Go to page action.

➤ **To test the Go to page action:**

1 At Reader level, select Advanced user.

2 Click Continue.

If the Contents page appears, your action sequence is working and you are ready to return to Author level and continue with the next example.

If your results were different than those described above, your action sequence is not working. Return to Author level and review the instructions in Example 1.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

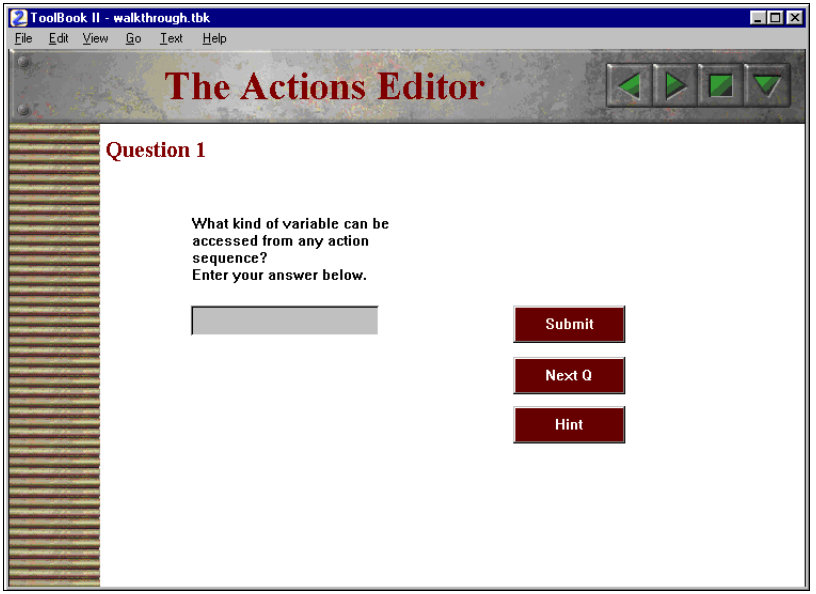
Glossary

Index

Example 2: Using variables

There are many ways you can use variables in an action sequence. In this example, a global variable stores the number of times a user clicks a Hint button. The action sequence uses this number to determine which hint to display.

At Author level, open page six of the Actions Editor Walkthrough book (named Question 1). There are several interactive objects on this page, including a fill-in-the-blank question object and three buttons. One of the buttons is labeled Hint. You will create an action sequence for this button that displays text as "hints" to the user.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Figure 2 shows the action sequence for Example 2 and the Variables dialog box. Instructor executes the action sequence as follows: First, Instructor evaluates each conditional expression in turn, comparing the value stored in the global variable named g_HintCounter against a specific integer. If an expression is true (for example, if the value stored in g_HintCounter is 0), Instructor runs the Display Popup text action that follows. Finally, the Set variable action increases the value of g_HintCounter by 1.

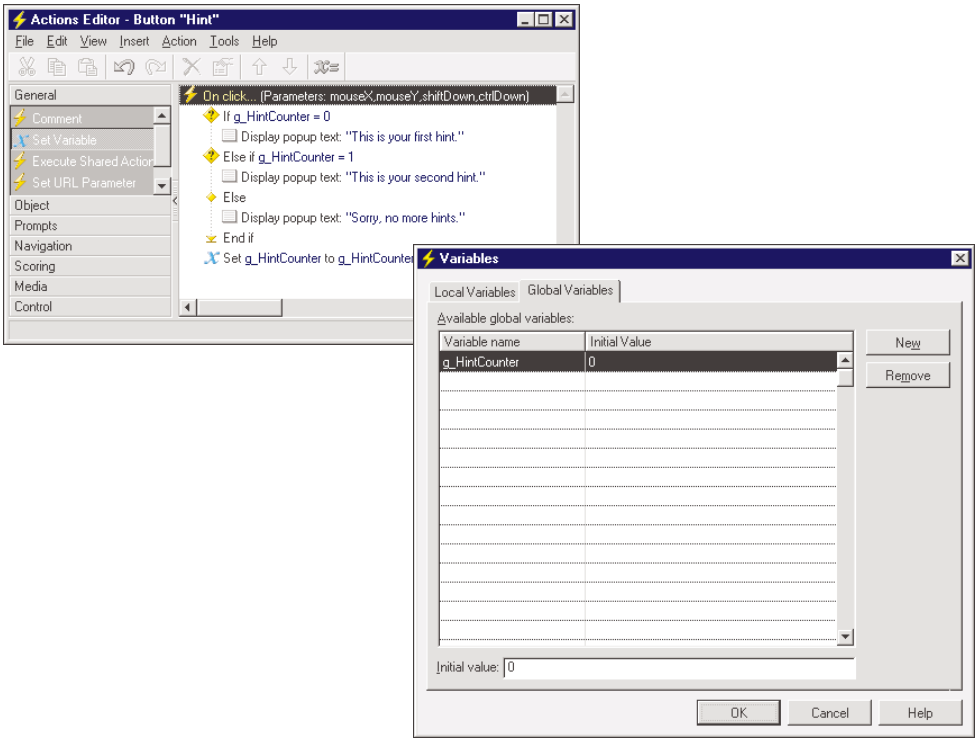


Figure 2
Full action sequence for Example 2

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Tip To make it easier to distinguish global variables from local variables, the names of global variables are often prefixed with an identifer such as g_.

Building the action sequence

Building this action sequence shows you how to:

- select an event.
- define a variable and change its value.
- define conditions for behavior.
- display text messages.
- cut, paste, and edit actions in the Actions Editor.

Opening the Actions Editor and selecting an event

Every action sequence for an object handles a specific event. The action sequence you create in this example occurs in response to a user clicking the Hint button.

➤ **To open the Actions Editor and select an event:**

1. Select the Hint button.
2. From the Object menu, choose Actions.

   The Actions Editor appears.
3. If On click is not selected, click the name of the event and select On click from the Event list.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Defining a variable

This action sequence uses a global variable named g_HintCounter to track how many times the user clicks the Hint button. This action sequence uses a global variable rather than a local variable since the value of a local variable is reset each time the action sequence runs.

➤ **To define a variable:**



1 Click the Define Variables button on the Actions Editor toolbar.

   The Variables dialog box appears.

2 On the Global Variables tab, click New.

3 In the Variables dialog box, type *g_HintCounter,* and then click OK.

4 In the Initial Value box, type *0* (zero).

5 Click OK.



Note When you define variables, they are not displayed as part of the action sequence. To see a list of variables defined for an action sequence, click the Define Variables button on the Actions Editor toolbar.

Inserting actions

In order to display a different hint each time a user clicks the Hint button, you need to track the number of times the user clicks this button. Since the value of the variable g_HintCounter corresponds with the number of clicks, you need to determine the value of g_HintCounter before displaying a hint. You can insert a condition to create a conditional expression that tests the value of g_HintCounter.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To insert a condition:**

1 From the Insert menu, point to Condition, and then choose If.

2 If the Properties for Condition dialog box does not open automatically, click the Properties button on the Actions Editor toolbar.

3 You can type *g_HintCounter = 0*, or you can use the lists in the Build Expression dialog box to create the same expression:

   A Click the Build expression button to open the Build Expression dialog box; in the Select the type of item to insert list, select Variables and parameters; in the Select a variable or parameter list, select g_HintCounter; click Insert into Expression.

   B Position the insertion point after g_HintCounter in the expression box, and then type *= 0*.

   C Click OK.

4 Click OK to close the Properties for Condition dialog box.

5 Using the skills learned above, insert two more conditions:

```
Else if g_HintCounter = 1
Else
```

To display a hint to a user, you need to show text on the page. This action sequence uses popup text to display a hint to a user.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To display popup text:

1. Select the first condition in the action sequence (If g_HintCounter = 0).
2. From the Insert menu, point to Prompts, and then choose Display Popup Text.
3. If the Properties for Action dialog box does not open automatically, click the Properties button on the Actions Editor toolbar.
4. Select Enter literal text.
5. In the Specify the text to display box, type *This is your first hint.*
6. Click OK to close the Properties dialog box.

To save time and steps, you can use the editing tools in the Actions Editor to copy and paste the Display Popup text action you just created.

➤ To copy and paste an action:

1. Select the action you just created.
2. From the Edit menu, choose Copy.
3. Select the second condition in the action sequence (Else if g_HintCounter = 1), and then from the Edit menu, choose Paste.

   A copy of the action appears below the second condition.
4. Select the third condition (Else) and paste again.

You will need to edit the text that is displayed by the actions you just pasted.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ To edit an action:



1. Select the first pasted action, and then click the Action Properties button on the Actions Editor toolbar.
2. Select the text in the Specify the text to display box, replace it by typing *This is your second hint*, and then click OK.
3. Select the second pasted action, replace its text with *Sorry, no more hints are available for this question,* and then click OK.

Now that you have created a conditional expression that tests the value of the g_HintCounter variable, and a Display Popup Text action that runs if the conditional expression is true, you need to insert an action that increments the value of g_HintCounter. You can use the Set variable action to change the value of a variable.

➤ To set the value of a variable:

1. In the Actions Editor, select the last line of the action sequence (End if).
2. From the Insert menu, point to Action, point to General, and then choose Set Variable.

   The Properties for Action dialog box appears.
3. In the Choose a variable to set list, select g_HintCounter.
4. In the Set the variable to this value box, type *g_HintCounter + 1*, and then click OK.
5. In the Actions Editor, from the File menu, choose Update Actions & Close.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Testing the action sequence

You can test to see whether this action sequence executes properly at Reader level.

➤ **To test an action sequence:**

1 Press F3 to switch to Reader level.

2 Click Hint a few times.

The first three times you click Hint, a different hint should display If you see "This is your first hint," followed by "This is your second hint," and finally, "Sorry, no more hints are available for this question," your action sequence is working. You can now return to Author level to complete the next example or exit the book.

If your results do not match the description above, your action sequence is not working. Return to Author level and review the steps in this example.

Example 3: Changing object properties

One advantage of online learning is that users can move through a course at their own pace and choose which topics to visit based on their individual information needs. You can help users track their progress by creating a table that is continually updated to show users the pages they have visited. In this example, you will change the *checked* property of a table item depending on whether its corresponding page has been visited.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

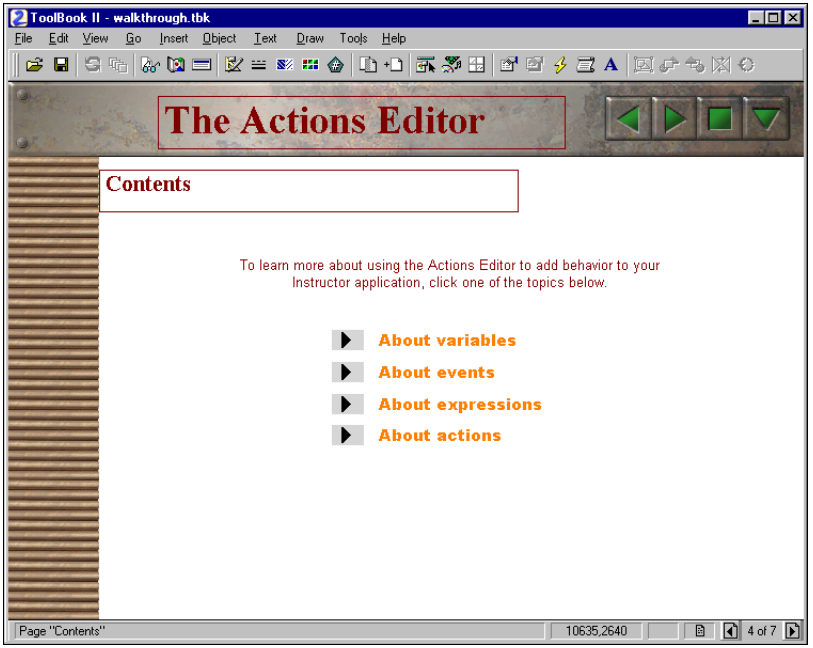
PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

At Author level, open page four (named Contents) of the Actions Editor Walkthrough book. You will see a list of page titles. Although no check boxes are visible, each of the titles in this list is a separate check box object.



Select the object labeled About variables and click the Properties button on the toolbar. On the Graphics tab, note that there are two graphics assigned to the button: one for its normal state and one for its checked state. You will create an action sequence for this object that determines whether the page About variables has been visited and, if it has, changes the state of the button to checked.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Figure 3 shows the action sequence for Example 3. Instructor executes the action sequence as follows: First, Instructor checks the visited property of page About variables. If this property is equal to true, the user has visited the page. Instructor then changes the state of the About variables button to checked by setting its checked property to true. If the page has not been visited, Instructor does not change the state of the button.

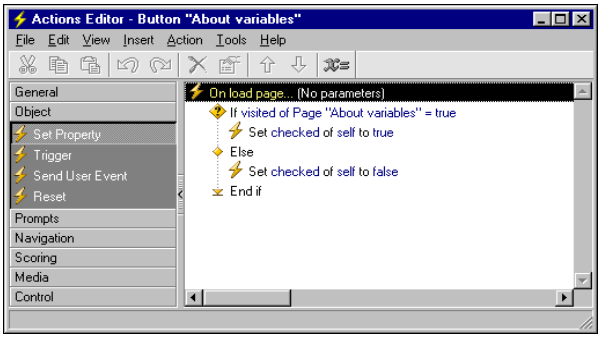


Figure 3
Full action sequence for Example 3

Building the action sequence

Building this action sequence shows you how to:

- select an event.
- define conditions for behavior.
- change the value of an object property.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Opening the Actions Editor and selecting an event

Because you want users to view an updated list each time they open the Contents page, you will want the action sequence to run each time the Contents page loads. The event On load page corresponds with the user opening a page in the book.

➤ **To open the Actions Editor and select an event:**

1 On the Contents page, select the check box object labeled About variables.

2 From the Object menu, choose Actions.

The Actions Editor appears.

3 In the Event list, select On load page.

Inserting actions

The first task in this action sequence is to determine whether or not a user has visited the page named About variables. You can insert a condition to create a conditional expression that tests whether a page has been visited.

➤ **To insert a condition:**

1 From the Insert menu, point to Condition, and then choose If.

The Properties for Condition dialog box appears.

2 In the Condition box, type *visited of Page "About variables" = True,* and then click OK.

3 From the Insert menu, point to Condition, and then choose Else.

To change the state of the About variables button from unchecked to checked, you can use the Set property action.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set an object property:**

1 Select the first condition in the action sequence (If visited of Page "About variables").

2 From the Insert menu, point to Action, point to Object, and then choose Set Property.

The Properties for Action dialog box appears.

3 Do the following:

A In the Select the object for which to set the property list, select self.

B In the Select the property to set list, select checked.

C Check Specify the new value for the property as a literal value.

D In the Select logical value list, select true, and then click OK.

4 Select the Else condition.

5 Following the steps above, create an action that sets the checked property of self to false.

6 Click OK.

7 From the File menu, choose Update Actions & Close.



Tip If you want to use the list items to send users to pages, you can follow the steps in Example 1 to create an action sequence that sends users to specific pages in response to the On click event.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Testing the action sequence

You will want to test this action sequence to ensure that the state of the button changes after the About variables page is visited.

➤ **To test the action sequence**

1. Press F3 to switch to Reader level.
2. Using the navigation buttons, go to the About variables page of the book and then return to the Contents page.

   If the check graphic appears next to About variables in the Contents list, your action sequence is working, and you are ready to return to Reader level and continue with the next example.

   If the graphic doesn't change, your action sequence is not working. Return to Author level and review the steps above.

Example 4: Using shared action sequences

You can use a shared action sequence any time you want to reuse the same sequence of behaviors. You can also use a shared action sequence to perform a series of actions that returns a value. In this example, a shared action sequence is used in a program that calculates the amount you would pay for sales tax in different cities.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

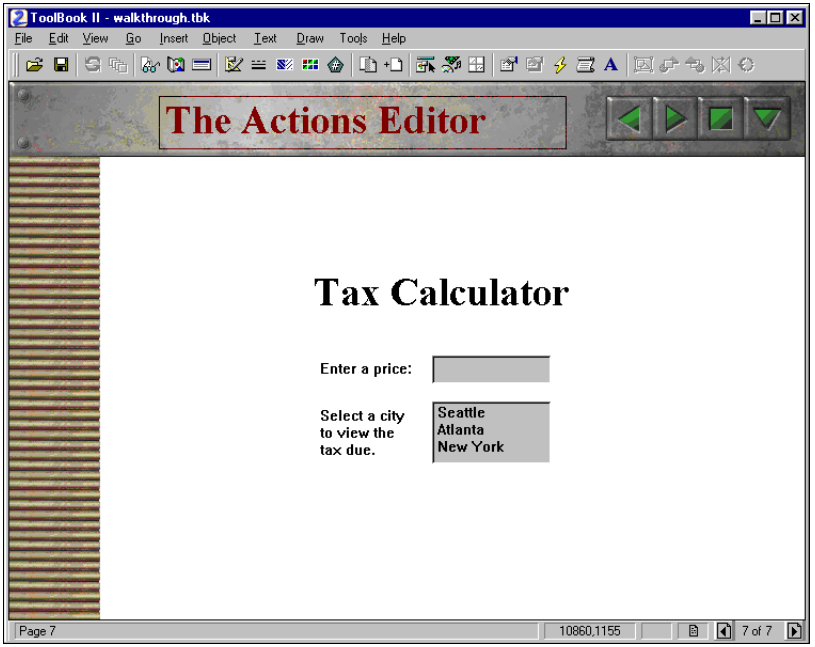
PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

At Author level, open the final page of the Actions Editor Walkthrough book titled "Tax Calculator." You will see two interactive objects on the page: A text field for entering a price and a list box containing three city names—Seattle, Atlanta, and New York.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Figure 4 shows the action sequence for Example 4. Instructor executes the action sequence as follows: First, Instructor evaluates which city the user selected from the list box. Then, it executes the shared action sequence, sending the appropriate tax rate and the text entered in the Enter a price field as parameters. The shared action sequence uses these parameters to calculate and return a value to the action sequence that called it. The value returned by the shared action sequence is stored in a local variable named Tax. Finally, Instructor displays the value of the local variable (the results of the calculation) to the user.

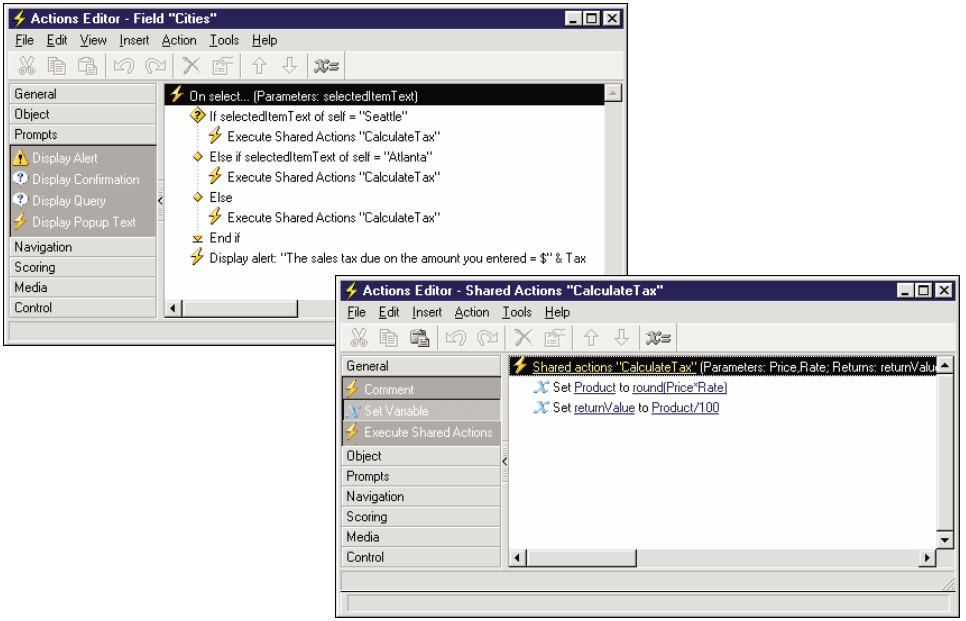


Figure 4
The full action sequence and shared action sequence for Example 4

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Building the shared action sequence

Building this shared action sequence shows you how to:

- create a new shared action sequence.
- define parameters and variables for a shared action sequence.
- calculate a value and store it in a local variable.
- return a value to an action sequence.

Creating a new shared action sequence

Shared action sequences are stored by unique names as properties of your book. You access the Shared Actions dialog box to create, edit, duplicate, import, or export any shared action sequence.

➤ **To create a new shared action sequence:**

1. From the Object menu, choose Shared Actions.

   The Shared Actions dialog box appears.

2. Click New.

   The Actions Editor appears.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Defining parameters and variables

When you create a shared action sequence, you can define the parameters it uses to complete any operations it performs. In this example, the parameters required to calculate sales tax are a price and a sales tax rate.

➤ **To define a parameter:**



1 Click the Define Variables and Parameters button on the Actions Editor toolbar.

   The Parameters and Variables dialog box appears.

2 On the Parameters tab, click New.

3 In the Parameter dialog box, enter *Price* as the name, and then click OK.

4 Repeat the steps above to define a second parameter and name it *Rate*.

It is sometimes easier to divide a long equation into two parts. You can use a variable to store the result of one calculation. Then, you can use that variable in a second calculation.

➤ **To define a variable:**

1 On the Local Variables tab, click New.

2 In the Name box, select NewVariable, and in its place, type *Product.*

3 In the Initial value box, type 0 (zero).

4 Click OK.

Inserting actions

You can easily complete the calculation by dividing it into two steps. First you will multiply the parameters Rate and Tax, round the result to the nearest integer, and store the resulting value in the local variable named Product.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Then you will use a variable named returnValue to store the final value that results from the operations that are carried out in the shared action sequence.

➤ **To set the value of a variable:**

1 From the Insert menu, point to Action, point to General, and then choose Set Variable.

The Properties for Action dialog box appears.

2 In the Choose a variable to set list, select Product.



3 Click the Build expression button to open the Build Expression dialog box, and then do the following:

A In the Select the type of item to insert list, select Operators and functions; in the Select a category list, select All functions; in the Select an operator or function list, select round; click Insert into Expression.

B In the expression box, insert the cursor between the parentheses after round.

C In the Select the type of item to insert list, select Variables and parameters; in the Select a variable or parameter list, select Price; click Insert into Expression.

D In the Select the type of item to insert list, select Operators and functions; in the Select a category list, select Math operators and functions; in the Select an operator or function list, select the multiplier symbol (*); click Insert into Expression.

E In the Select the type of item to insert list, select Variables and parameters; in the Select the variable or parameter list, select Rate; click Insert into Expression.

4 Click OK to close the Build Expression dialog box, and then click OK to close the Properties dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

To return the value calculated in the shared action sequence, you use the Set Return Value action. You can change the form of the result to display as a dollar amount limited to two decimals, by setting the variable returnValue to the value of Product divided by 100.

➤ **To set the return value:**

1 From the Insert menu, point to Action, point to General, and then choose Set Return Value.

The Properties for Action dialog box appears.



2 Click the Build expression button to open the Build Expression dialog box, and then do the following:

A In the Select the type of item to insert list, select Variables and Parameters; in the Select a variable or parameter list, select Product; click Insert into Expression.

B In the Select the type of item to insert list, select Operators and functions; in the Select a category list, select Math operators and functions; in the Select an operator or function list, select the division symbol (/); click Insert into Expression.

C In the expression box, insert your cursor after the division symbol and type *100*.

3 Click OK to close the Build Expression dialog box, and then click OK to close the Properties dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Saving and naming the shared action sequence

Once you have built a shared action sequence, you need to save it under a unique name; then, you can call it from any other action sequence.

➤ **To save a shared action sequence:**

1. In the Actions Editor, from File menu, choose Update Actions & Close.
2. In the Shared Actions dialog box, in the Name box, select NewSharedAction, and then type CalculateTax.
3. Click Close.

Building the action sequence for the list box

The shared action sequence you just created runs only when called by another action sequence. Now you will create an action sequence for the list box object named Cities. The action sequence includes an action that calls CalculateTax and supplies it with the two parameters it needs to complete its operation: Price and Rate.

Building this action sequence shows you how to:

- select an event.
- define a variable to store the value returned by the shared action sequence.
- define conditions for behavior.
- call and supply parameters to a shared action sequence.
- display a message in a dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Opening the Actions Editor and selecting an event

The event that activates this shared action sequence is a user selecting a city from the Cities list box. You need to create an action sequence for the list box that runs in response to the On select event.

➤ **To open the Actions Editor and select an event:**

1. On the Tax Calculator page of the Actions Editor Walkthrough book, select the Cities list box.
2. From the Object menu, choose Actions.

   The Actions Editor appears.
3. In the Event list, choose On select.

Defining variables

The shared action sequence you created returns a value. You need to define a local variable to store that value while the action sequence runs.

➤ **To define a local variable:**



1. Click the Define Variables button on the Actions Editor toolbar.
2. On the Local Variables tab, click New.
3. In the Variables dialog box, type *Tax,* and then click OK.
4. In the Initial value box, type 0 (zero).
5. Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Inserting conditions to determine list box selection

Depending on which city a user selects in the list box, you will send a different tax rate as a parameter to the shared action sequence. So, first you need to determine the user's selection. You can insert a condition to create a conditional expression that tests a user's selection.

➤ **To insert a condition:**

1 From the Insert menu, point to Condition, and then choose If.

The Properties for Condition dialog box appears.



2 Click the Build expression button to open the Build Expression dialog box.

3 In the Build Expression dialog box, do the following:

A In the Select the type of item to insert list, select Object properties; in the Select an object list, select self; in the Select a property list, select selectedItemText; click Insert into Expression.

B Position the insertion point at the end of the expression, and then type = *"Seattle"*.

4 Click OK to close the Edit Expression dialog box, and then click OK to close the Properties dialog box.

5 Repeat the above steps until you have created a conditional expression that tests for each item in the list.

```
If selectedItemText of self = "Seattle"
Else if selectedItemText of self = "Atlanta"
Else
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Executing a shared action sequence

To perform the calculation, you need to call the shared action sequence you created above. You use a specific action—Execute shared actions—to run the shared action sequence. When you run the shared action sequence, you need to supply values for the two parameters it requires: Rate and Price.

➤ **To execute a shared action sequence:**

1. Select the first condition in the action sequence (If selectedItemText of self = "Seattle").
2. From the Insert menu, point to Action, point to General, and then choose Execute Shared Actions.

   The Properties for Action dialog box appears.
3. In the Select a shared action sequence list, select CalculateTax.

   The two parameters you defined when you created the shared action sequence—Price and Rate—appear under Parameter Name.



4. Under Parameter Value, insert the cursor in the box next to Price, and then click the Build expression button.
5. In the Build Expression dialog box, do the following:
   - **A** In the Select the type of item to insert list, select Object properties.
   - **B** In the Select an object list, select Field "Price."
   - **C** In the Select a property list, select text, and then click Insert into Expression.
6. Click OK.
7. Under Parameter Value, insert the cursor in the box next to Rate, and then type *8.6*.
8. In the Specify a variable to store the return value list, select Tax.
9. Click OK.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

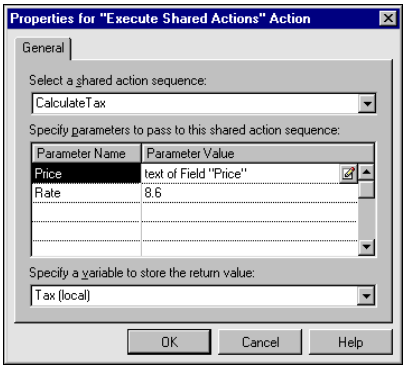


Figure 5
Properties for "Execute Shared Actions" Action

Copying, pasting, and editing an action

There is only one difference between the action you just inserted and the actions you will insert after the next two conditions: the value of Rate. When actions are very similar, you will save time by copying, pasting, and editing the first action you create.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To copy, paste, and edit an action:**

1 Select the action you inserted in the steps above.

2 From the Edit menu, choose Copy.

3 Select the second condition in the Action sequence (Else if selectedItemText of self = “Atlanta”).

4 From the Edit menu, choose Paste.



5 While the action you pasted is selected, click the Properties button on the Actions Editor toolbar.

6 In the Properties for Action dialog box, select the parameter value next to Rate, and type *6*.

7 Repeat the steps above to insert the same action under the third condition in the action sequence (Else), select the value next to Rate, and then type *8.25*.

Inserting the Display alert action

So far, the actions you have inserted into the action sequence determine which item the user selects in the list box and call a shared action to perform a calculation from the information it supplies. Now you need to give feedback to the user displaying the dollar amount due in tax. You can use the Display alert action to show users a message in a dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

➤ **To display an alert:**

1 In the Actions Editor, select End if.

2 From the Insert menu, point to Action, point to Prompts, and choose Display Alert.

The Properties for Action dialog box appears.

3 Select Enter an expression.



4 In the Specify the text to display box, click the Build expression button.

5 In the Build Expression dialog box, build the following expression: "The sales tax due on the amount you entered = $" &Tax

- **A** Type *"The sales tax due on the amount you entered = $"*
- **B** Type & (ampersand).
- **C** In the Select the type of item to insert list, select Variables and parameters.
- **D** In the Select a variable or parameter list, select Tax, and then click Insert into Expression.

6 Click OK to close the Build Expression dialog box, and then click OK to close the Properties dialog box.

7 In the Actions Editor, from the File menu, choose Update Actions & Close.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Testing the action sequence

The following steps test both action sequences you created in this example.

➤ **To test the action sequence:**

1. Press F3 to switch to Reader level.
2. In the Price field, type *580*.
3. In the list box, click each of the cities and note the message that Instructor displays.
   - When you click Seattle, a dialog box should appear that reads, "The sales tax due on the amount you entered = $49.88."
   - When you click Atlanta, a dialog box should appear that reads, "The sales tax due on the amount you entered = $34.8."
   - When you click New York, a dialog box should appear that reads, "The sales tax due on the amount you entered = $47.85."

If your results do not match those listed above, return to Author level and review the instructions. If they do match, you have successfully completed the final example.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Troubleshooting action sequences

If your action sequences aren't producing the expected behavior, review them for the following common errors:

- **Typing a name in the Build Expression dialog box that doesn't exactly match the name of the object to which it refers** For example, did you type "about varailbes" instead of "about variables"? A typo is often difficult to pinpoint as the source of a faulty action sequence. You can avoid making typing mistakes by selecting object names and variables in the lists in the Build Expression dialog box.
- **Selecting the wrong event in the Event list** By default, when you open the Actions Editor, it opens to the most common event in its list of supported events (usually, On click). To create an action sequence for any other event, you must select that event in the Event list.
- **Providing insufficient information** Certain dialog boxes, including those for Display Confirmation and Display Query, have two tabs on which you need to select options. Your action sequence will not run properly if you do not enter all necessary information.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

15 Using the Actions Editor

16 Creating action sequences: Practical examples

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

PART V

Using advanced features

In this part of the *ToolBook II Instructor User Guide,* you will learn about several powerful features that you can use to further customize and extend your Instructor application. These chapters discuss how to import and export text, graphics, and other data; create and modify windows and menu bars; and add custom features to your application using technologies such as ActiveX, OLE, and ADO. This part also contains a chapter describing how to make native ToolBook applications run as quickly and efficiently as possible.

IN THIS PART

Chapter 17 Using resources

Chapter 18 Importing and exporting data

Chapter 19 Creating windows with viewers

Chapter 20 Creating and modifying menu bars

Chapter 21 Using ActiveX, Automation, and other Windows technologies

Chapter 22 Optimizing Instructor applications



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Chapter 17

Using resources

Resources are files that can be reused in your ToolBook II Instructor application. Some resources are used to create a cohesive interface for your application—cursors, icons, color palettes, and menus, for example. Another type of resource, a shared script, is a script that can be attached to many objects. Instructor also uses the resource system to manage graphical elements you add to your book, allowing you to use a graphic several times in your book with only minimal impact on the file size of your application. This chapter describes how to add, modify, and export resources.

IN THIS CHAPTER

About resources **446**

Creating and modifying resources **449**

Adding resources to a book **452**

Working with resources **454**

Getting information about a resource **461**

Replacing resources **463**

Exporting resources **465**

Removing resources from a book **466**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

About resources

A resource is a file that you can use multiple times within your ToolBook II Instructor application. You can use resources to:

- apply a bitmap to a button to make your interface more graphical and visually consistent with other Windows applications. For example, you can apply a bitmap of an arrow to a button that navigates to the next page.
- substitute a custom cursor for a ToolBook II system cursor.
- change the default drag image or no-drop cursor that appears when an object is dragged or dropped.
- add an icon that appears when your application is minimized.
- add a menu bar that appears in multiple windows of your application.
- apply a color palette to a book to reduce palette flashes that occur when displaying bitmaps on 256-color display devices.
- identify a shared script that is used by multiple objects in your application.
- substitute a custom Windows icon when you save your application as an EXE file.

When you add a resource to a book, Instructor places a copy of it in the Resource Manager, which is a common library of resources in the book. The resource is then available to all objects in the book that can use that type of resource; however, the resource does not appear in the book until you assign it to an object as a property.

An object uses a resource by referring to the stored copy by name or ID number. Instructor stores only one copy of the resource in the book, no matter how many objects refer to it. As a result, you can apply the resource to as many objects as you want without adding to the file size of your application.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

For example, if your book has multiple backgrounds with the same navigation buttons on every page, you would use a bitmap resource for the graphic image on the navigation buttons. Because only one copy of the bitmap is stored, your book's file size is much smaller than it would be if you duplicated the bitmap for each button.

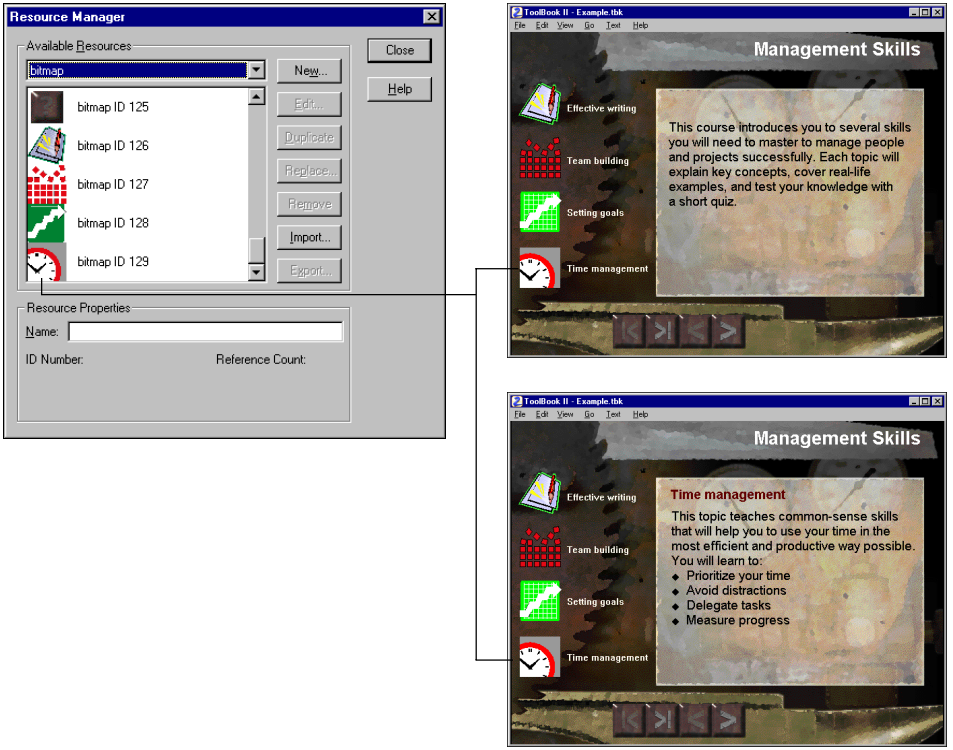


Figure 1
Resources are stored once but can appear often

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Note You cannot share the same copy of a resource among multiple books. You must import a separate copy of the resource for each book.

Customizing the Instructor interface using resources

To do this...	Assign this resource type...
Create a graphic button	Bitmap, cursor, icon
Apply a color palette to a book	Color palette
Change the drag image and no-drop cursor	Bitmap, cursor, icon
Apply a custom menu	Menu bar*
Change a viewer's† Minimize icon	Icon
Change a book's system cursor	Cursor, icon

* For details about creating custom menus, see Chapter 20, "Creating and modifying menu bars."

† For details about creating viewers, see Chapter 19, "Creating windows with viewers."

Tip If you plan to save your book as an EXE file, you can customize its Windows icon. Before you save your book as an EXE file, right-click on the title bar, choose Choose icon, and then, in the Choose Icon dialog box, import your custom icon or create a new icon. When you save your book as an EXE file, the title bar icon appears as the book's Windows icon.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Creating and modifying resources

You can create and modify bitmap, cursor, icon, menu bar, color palette, and shared-script resources using the utility applications that come with Instructor. Some resources can be created using other Windows applications; for example, you could create all your graphic images using Adobe® Photoshop® and then import them to Instructor.

Menu bar and shared-script resources are unique to Instructor. You create and use them only in ToolBook II products.

Creating and modifying resources

To create this resource type...	Identified by this file extension...	Use this...
Bitmap	.bmp, .dib .gif, .jpg	ToolBook II BitEdit or another Windows program
Cursor	.cur*	ToolBook II Icon/Cursor Editor or another Windows program
Font	.ttf	TrueType™ fonts
Icon	.ico*	ToolBook II Icon/Cursor Editor or another Windows program
Menu bar	.mnu	ToolBook II Menu Bar Editor
Palette	.pal†	ToolBook II PalEdit or another Windows program
Shared script	—	Instructor's script editor

* These files may contain more than one image. Instructor uses the first image it encounters in the file.

† Instructor can extract color palette information from Windows Metafile Format (WMF) and device-independent bitmap (DIB) files when you specify them in the Resource Manager dialog box or import them using OpenScript.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

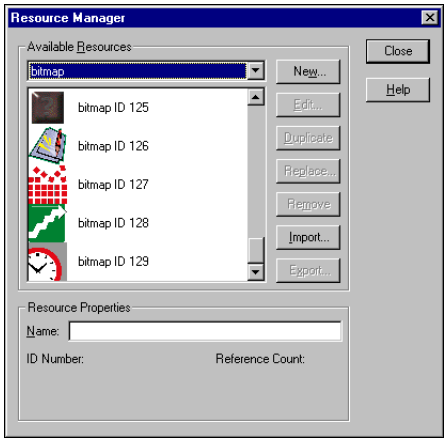
You can access ToolBook II utility applications for creating and modifying resources from the Resource Manager dialog box (the appropriate utility opens when you select a resource to create or edit) or from the Tools menu. To open one of the ToolBook II utilities, from the Tools menu, point to Applications, and then choose an application from the submenu.

Detailed information about using each tool is available from online Help.

➤ **To create or modify a resource in the Resource Manager:**

1 From the Object menu, choose Resources.

The Resource Manager dialog box appears.



2 In the Available resources list, choose the type of resource you want to create or modify.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

3 Do one of the following:

- If you're creating a new resource, click New.
- If you're editing an existing resource, select the resource from the list, and then click Edit.

A utility appears, in which you can create or modify the resource. Save the resource, and then close the utility.

4 In the Name box, if necessary, type a unique name for the resource, and then click Close.

You can also duplicate resources that you have made available to a book. For example, if you have an existing cursor resource that you want to use in a slightly modified form, you can duplicate the existing resource, edit the copy, and give it a unique name.

➤ **To duplicate a resource:**

1 From the Object menu, choose Resources.

The Resource Manager dialog box appears.

2 In the Available resources list, select the type of resource you want to duplicate.

3 In the graphical list of resources, select the resource you want to duplicate.

4 Click Duplicate.

Instructor duplicates the resource.

5 *Optional.* Select the new resource, and then click Edit to modify it.

6 Click Close to close the Resource Manager dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Adding resources to a book

There are many ways to add a resource to your book. You can:

- create a new resource and add it to your book.
- import an existing file to be used as a resource.
- copy a resource from another book.
- add an object that includes a resource from one of the ToolBook II Catalogs. Instructor automatically imports the resource into the current book.
- use OpenScript to refer to a resource in another book. Instructor automatically imports the resource into the current book.
- insert an inline graphic into a field or record field.
- import an Instructor page or an entire book that contains resources.

If you've planned in advance all the artwork, system elements, and other resources that you want to include, you can use the Resource Manager to add them to the book all at once. If you're designing as you go, or if you discover that you need additional resources, you can add individual resources one at a time.

➤ To add a resource to a book using the Resource Manager:

Using the interface

1. From the Object menu, choose Resources.

   The Resource Manager dialog box appears.

2. In the Available resources list, select the type of resource you want to add.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

3 Do one of the following:

- To import a file as a resource, click Import.

  The Import Resource dialog box appears. Select the resource you want to import, and then click OK.

- To create a new resource, click New.

  The ToolBook II utility application for creating the type of resource you've selected opens. Create the resource using the utility application. When you save the resource, you will have the option to save it as a resource in your book. Do this, and then close the utility application.

The resource appears in the graphical list in the Resource Manager dialog box.

4 Type a unique name in the Name box under Resource properties.

Instructor assigns an ID number to the resource that you can also use to refer to it.

5 Close the Resource Manager dialog box.

Using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

- To import a resource, use the *import* command:

```
import icon resource "c:\toolbook\icon\plane.ico" as "plane"
```

- To copy a resource from another Instructor book, use the *copy resource* command:

```
copy resource icon "airplane" of book "travel.tbk" to this book
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

- To automatically import a resource from another book, set a resource property to a resource that is owned by a different book:

```
dragImage of button "Play MIDI" = cursor "hand.cur" \
   of book "tutor.tbk"
```

Assigning resources as object properties

You can assign a resource that you have already added to your book to an individual object, or you can create a resource for an object when you are ready to assign it. Resources are considered properties of an object. You can assign a resource to an object using the Properties dialog box.

The following section, "Working with resources," describes how to assign a resource as an object property.

Working with resources

The steps you take to assign a resource as an object property differ depending on the type of object and the type of resource. This section describes how to assign various resources to different objects.

Adding a graphic to a button

You can assign as many as four different graphics to a button: one for the button's normal state, one for its inverted state (when the button is clicked), one for its disabled state (when the button cannot be clicked), and one for its checked state (when an option is chosen).



Note You can also assign a rollover graphic to a button that will display when the mouse pointer pauses over the button. On the Graphics tab in the Properties for Button dialog box, select the option labeled Use checked graphic as rollover graphic.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ To add a graphic to a button:

Using the interface



1. Select the button to which you want to add the graphic, and then click the Properties button on the Instructor toolbar.
2. In the Properties for Button dialog box, click the Graphics tab.
3. Under Button graphics, choose a button state (Normal, Invert, Disabled, or Checked), and then click Choose Graphic.

   The Choose Graphic dialog box appears.
4. Select the type of resource you want (bitmap, cursor, or icon) in the Available resources list.

   The resources listed in the Choose Graphic dialog box include all the resources of the selected type that have been imported into the current book. Any new resources you import are added to the book's resource list.
5. Do one of the following:
   - To use an existing resource, select the graphic you want in the graphical list of resources.
   - To import a new resource, click Import. In the Import Graphic or Import Resource dialog box, locate the graphic file you want and click OK; then select the resource in the graphical list of resources.
6. Click OK to close the Choose Graphic dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

7 *Optional.* Under Graphic options, do one or more of the following:

- If you want the bitmap to fill the button, select Stretch the graphic to fit the button. (You cannot stretch icons or cursors because they have a fixed size.)
- If you want the button to size itself to the graphic, select Size the button to fit the graphic.
- If you want to use the Checked graphic as the rollover graphic (the graphic you see when the pointer is paused over a button), select Use checked graphic as rollover graphic.

8 Repeat steps 3–7 to assign graphic resources for the other button states.

9 *Optional.* Close the Properties for Button dialog box.

You might need to resize the button to see the graphic.

Using OpenScript

- To apply a graphic to a button, set the *normalGraphic, invertGraphic, disabledGraphic, or checkedGraphic* property of a button to a bitmap, cursor, or icon resource:

```
normalGraphic of button "Baseball" = icon "normalBaseball"
```



Note To remove a graphic, on the Graphics tab of the Button Properties dialog box, choose the button state, and then click Clear Graphic.

Applying a color palette to a book

If your users are likely to run their computers in 256-color mode, you'll want to be aware of how Windows and Instructor handle color palettes. For details about color and palettes, see Chapter 9, "Using graphics."



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

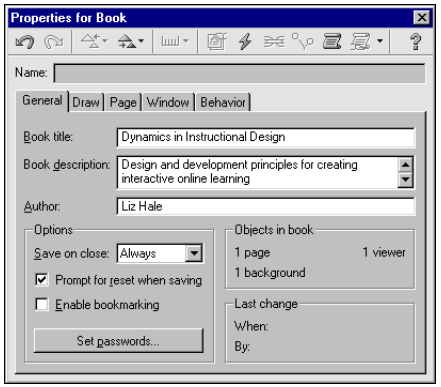
If you have graphics that use a different set of colors on successive pages in 256-color mode, there may be a momentary palette flash as the user navigates from page to page and Windows switches from the old palette to the new one. You can eliminate palette flashes in your book by assigning a color palette. You can assign a color palette to a book and to an individual graphic. A book palette takes priority over any graphic object's individual palette. The advantage of assigning a book palette is that Instructor uses it to display all graphics, thus eliminating palette flashing as the user turns pages or selects different objects. The disadvantage is that every graphic must use the same color palette, which may not accurately match all the colors in your imported graphics.

➤ To assign a color palette to a book:

Using the interface

1 From the Object menu, choose Properties for Book.

The Properties for Book dialog box appears.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

2 On the Draw tab, under Palette, click Choose.

The Choose Palette dialog box appears.

3 Do one of the following:

- To assign an existing palette, select the palette you want in the graphical list of resources.
- To import a palette, click Import to display the Import Resource dialog box, navigate to and select the palette file (with a .pal file extension) that you want, and then click OK.

4 Click OK to close the Choose Palette dialog box.

Using OpenScript

- To assign a color palette to a book, set the book's *palette* property to the name or ID of a palette resource you imported earlier. You can also import a new palette directly by setting the book's *palette* property to the palette file. For example:

```
palette of this book = palette "myCustomPalette"
--Imports the palette as a resource as well as assigning it
palette of this book = "c:\toolbook\custom.pal"
```

Changing the drag image and no-drop cursor

You can change the cursor or drag image that appears when a user drags an object, as well as the no-drop cursor that appears when a user drags the object into an area where it cannot be dropped. Use Instructor's default drag-and-drop cursors, or customize your book by assigning unique resources.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ To change the drag image and no-drop cursor of an object:

Using the interface



1. Select the object, and then click the Properties button on the Instructor toolbar.

   The Properties dialog box appears.

2. On the Drag & Drop tab, select either Can be dragged or Accepts dropped objects, and then click Change Graphic.

   The Choose Graphic dialog box appears.

3. Select the type of resource (bitmap, cursor, or icon) that you want to assign.

4. Do one of the following:

   - To assign an existing resource, select the resource you want in the graphical list.
   - To import a resource, click Import, navigate to and select a graphic file, and then click OK.

5. *Optional.* Close the Properties dialog box.

6. Switch to Reader level (press F3), and drag and drop the object to test the results.

Using OpenScript

- To change the cursor that appears when an object is dragged, set the object's *dragImage* property to a bitmap, cursor, or icon resource:

```
dragImage of object "baseball" = bitmap "blur"
```

- To change the cursor that indicates an object cannot be dropped, set the object's *noDropImage* property to a bitmap, cursor, or icon resource:

```
noDropImage of object "bleachers" = cursor "no drop"
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Changing a viewer's Minimize icon

You can change the icon that appears when a viewer is minimized using either the Properties dialog box or OpenScript. For more information about viewers, see Chapter 19, "Creating windows with viewers."

➤ **To change a viewer's Minimize icon:**

Using the interface

1. From the Object menu, choose Viewers.

   The Viewers dialog box appears.

2. In the Viewers of book list, choose the viewer for which you want to change the Minimize icon, and then click Properties.

   The Properties for Viewer dialog box appears.

3. On the Style tab, under Icon, click Choose icon.

4. In the Choose Icon dialog box, create or import an icon, and then click OK.

5. *Optional*. Close the Properties for Viewer dialog box.

Using OpenScript

- To change the icon that appears when a viewer is minimized, set the *icon* property to an icon resource:

```
icon of mainWindow = icon "fancy"
icon of viewer "Navigation" = icon "plain"
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Changing a book's system cursor with OpenScript

Instructor comes with a number of system cursors that you can use. Alternatively, you can import a cursor as a resource and then set the OpenScript *sysCursor* property to that resource. For example:

```
sysCursor = cursor "hand"
```

You can make the cursor change whenever it passes over something that you want the user to click. For example, use this script to make the cursor change to a pointing hand every time it passes over a button:

```
--Sent when the cursor crosses the bounds of an object
to handle mouseEnter
   if object of target is "button" then
      system oldCursor                --Declares the system variable
      oldCursor = sysCursor           --Saves the current cursor
      sysCursor = cursor "hand"       --Resets the cursor to a
                                      --resource (hand graphic)
   end if
end mouseEnter

to handle mouseLeave          --The mouse pointer is leaving the object
   system oldCursor           --Redeclares the variable
   sysCursor = oldCursor      --Resets the cursor to its old value
end
```

Getting information about a resource

You can get information about a resource (such as the size or ID number) from the Resource Manager dialog box or by requesting the *resourceInfo* property in the Command window.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To get information about a resource:**

Using the interface

1 From the Object menu, choose Resources.

The Resource Manager dialog box appears.

2 In the Available resources list, select the type of resource about which you want to view information.

3 In the graphical list of resources, select the resource.

Information about the selected resource appears under Resource properties.

Using OpenScript

◆ You can use the OpenScript *resourceInfo* property to retrieve information about a specific resource. For example:

```
request resourceInfo of icon "world"
```

The information Instructor displays varies depending on the resource type, as the following table shows. For all resource types, the first item of information that Instructor displays is the reference count, which is the number of times the resource is referenced within the book. The reference count also appears in the Resource Manager dialog box.



Tip The reference count is not updated for references to a resource in a script or in user properties. Therefore, it is possible for a resource to have a reference count of zero but still be used in a book.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Getting resource information with the *resourceInfo* property

Resource type	Information
Bitmap	Reference count; bitmap width in pixels; bitmap height in pixels; color depth as 1, 4, 8, 16, or 24 bit; bitmap size in bytes
Cursor	Reference count; icon width in pixels; icon height in pixels; color depth as 1 bit; the hotspot *x* and *y* position in pixels, which is the area that needs to be over an object when the mouse is clicked
Icon	Reference count, icon width in pixels, icon height in pixels, color depth as 1 or 4 bit
Menu bar	Reference count
Palette	Reference count, number of colors in the palette (up to 256)

Replacing resources

You can replace the resources in a book with other resources. For example, if you apply a bitmap resource to a button and then replace the resource with a new bitmap, the new bitmap appears in place of the old resource everywhere that it is used.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ To replace a resource in a book:

Using the interface

1. From the Object menu, choose Resources.

   The Resource Manager dialog box appears.

2. In the Available resources list, select the type of resource you want to replace.

3. In the graphical list of resources, select the resource you want to replace, and then click Replace.

   The Replace Resource or Import Graphic dialog box appears, depending on the type of resource you're replacing.

4. Navigate to and select the resource file you want to use, and then click OK.

5. *Optional.* Close the Resource Manager dialog box.

   The new resource replaces the old resource throughout the book.

Using OpenScript

- Use the *replace resource* command to replace an old resource with a new one. For the new resource, specify the file name of a resource of the same type. For example:

```
replace resource cursor "arrow" with "c:\cursors\arrow1.cur"
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Exporting resources

You can export icon and cursor resources that you have created or modified to use in other books or Windows programs.

➤ **To export a resource from a book:**

Using the interface

1 From the Object menu, choose Resources.

The Resource Manager dialog box appears.

2 In the Available resources list, select the type of resource you want to export.

3 In the graphical list of resources, select the resource you want to export.

4 Click Export.

The Export Resource dialog box appears.

5 Enter a name for the resource, navigate to and select a file to which to export it, and then click OK.

Instructor exports the resource.

Using OpenScript

◆ Use the *export* command to export a resource:

```
export resource cursor "point" as "c:\point.cur"
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Removing resources from a book

If you're concerned about file size, you can remove all unused resources from your book to reduce the book's file size. When you remove a resource, you are removing only the book's copy of the resource; the resource's original file is unaffected.

You cannot remove a resource from a book if it has been assigned to an object. You must first remove each object's reference to the resource using OpenScript or the object's Properties dialog box. For example, if you create a graphic button with a bitmap resource, you must remove the bitmap from the button before you can remove the resource from the book. The following table explains how to remove an object's reference to a resource by setting the object's property to the Instructor default.



Important The Resource Manager dialog box maintains a reference count, which reflects the number of times a resource is used in a book. When the reference count for a resource is zero, Instructor allows you to remove the resource from your book. However, this reference count is not updated for references to a resource within a script or in user properties. Be certain not to remove any resources that you intend to continue referencing in a script or in user properties.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Removing an object's reference to a resource

To remove this resource...	Do this...

Bitmap	Select the button, and then click the Properties button on the toolbar. On the Graphics tab, choose the button state, and then click Clear Graphic.
	Alternatively, type in the Command window or a script:
	`normalGraphic of button "Baseball" = null`
	`or clear normalGraphic of button "Baseball"`

Icon	Type in the Command window or a script: `icon of viewer "Statistics" = null`

Drag image or no-drop cursor	Type in the Command window or a script:
	`dragImage of button "Play MIDI" = null`
	`noDropImage of button "Stop MIDI" = null`

Color palette	Assign a different palette to the book by typing in the Command window or a script: `set palette of this book to palette ID 101`
Menu bar resource*	Type in the Command window or a script: `menuBar of viewer "Statistics" = null`

System cursor	Type in the Command window or a script:
	`sysCursor = 1`
	`or sysCursor = default`

* For details, see Chapter 20, "Creating and modifying menu bars."

After you have removed all references to a resource, you can remove the resource from a book.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ To remove a resource from a book:

Using the interface

1. From the Object menu, choose Resources.

   The Resource Manager dialog box appears.

2. In the Available resources list, select the type of resource you want to remove.

3. In the graphical list, select the resource you want to remove.

4. Click Remove, and then close the Resource Manager dialog box.

Using OpenScript

- Use the *remove resource* command to remove a resource from a book:

```
remove resource cursor ID 100
remove resource palette "gray tones" of book "blk_wht.tbk"
```

  The resource no longer appears in the Resource Manager dialog box for that book.

- Use the *resourceUsedBy()* function to locate all objects using a particular resource. For example:

```
request resourceUsedBy (bitmap "Next", this book)
```



Note If you delete an object that has a resource assigned as a property (for example, a button with a graphic resource assigned to a particular state), you cannot immediately remove the resource from the Resource Manager dialog box. Because Instructor saves a copy of the deleted object in memory to allow you to undo your action, you must first perform another action (such as clicking on another object) before you can delete the resource from the Resource Manager dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Chapter 18

Importing and exporting data

You can import and export text and graphics in ToolBook II Instructor. In addition, you can import books and pages from other Instructor and ToolBook II Assistant applications into your current Instructor application. Importing and exporting affords flexibility and saves time, because you can leverage your development efforts by reusing material. Further, you can incorporate material from a number of sources, adding variety and extending functionality. This chapter discusses how to import and export data and describes how to customize the import process to fit the needs of your application.

IN THIS CHAPTER

About importing and exporting **470**

Importing and exporting text **471**

Importing and exporting graphics **484**

Importing books and pages **486**

Customizing the import process **490**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

About importing and exporting

Importing is the process of reading information from a file and copying it into your current book. Exporting is the opposite: reading information from your current book and copying it to another file. In a ToolBook II Instructor application, you can import and export text and graphics. You can also import pages and books from other Instructor and ToolBook II Assistant applications.

You can import and export text in two common text formats: a plain-text format called ASCII (American Standard Code for Information Interchange) and RTF (Rich Text Format). You can import graphics in a wide variety of common file formats, and you can export to BMP (bitmap file format), GIF (Graphics Interchange Format), and JPEG (Joint Photographic Experts Group format). Importing and exporting text and graphics allows you to incorporate material from other sources into your Instructor application and reuse material from one ToolBook II application to another.

Importing books and pages from other ToolBook II applications is a way to take advantage of your earlier work. For example, if you have created a tutorial application in Assistant, you can import its pages into new tutorials that you create in Instructor.

Importing or exporting is also useful when you need to share information with an application that doesn't support Dynamic Data Exchange (DDE), object linking and embedding (OLE), or the calling of functions in a dynamic-link library (DLL). For example, you might need information from a database that doesn't run under Windows. You could export the data from the database into a text file and then import the text file into your book. You could also export text from your book into a text file that you can then import into a database, spreadsheet, or word-processing program.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

This chapter discusses importing and exporting data in Instructor but does not discuss exporting your Instructor application for use on the Internet. To learn about exporting your application to an Internet-ready file format, see Chapter 24, "Distributing applications on the Internet."

Importing and exporting text

You can import and export text into a field or record field. A text field is an Instructor object that contains text; a text field can reside on the foreground or background of a page. A record field resides on the background of a page and displays changing text consistently from page to page (data-entry forms, for example). For more information about text and record fields, see Chapter 8, "Working with text."

When you import to a text field, you move data from one source document into one text field. When you import to a record field, you move data from one source file into multiple record fields. Exporting from text fields or record fields moves data into one text file.

You can import a text file into record fields as a fixed-field file or a delimited-field file. In fixed-field files, the field size is predetermined and constant. In delimited-field files, the size can vary and is delimited by special characters that you define. You must insert the delimiter in the source file (the file from which you are importing the text) at each place where you want a new record field to begin.

Importing text files into a text field

You can import data from any application that creates text files and saves them in either the ASCII format or RTF (a word-processing program, for example). The ASCII format (also called the text-only format) allows for only text characters to be contained in a file. RTF files include characters and some formatting, such as font and font style.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Text pasted from the Windows Clipboard is generally formatted as RTF. For information about copying and pasting text using the Clipboard, see Chapter 8, "Working with text."

➤ **To import text into a text field:**

1. From the tool palette, choose the field tool, and then draw a field on your page.
2. Choose the select tool from the tool palette, and then select the field you have just drawn.
3. From the Insert menu, choose Text.

   The Import Text dialog box appears.
4. In the Import Text dialog box, navigate to and select the text file you want to import, and then click OK.

   The text from the source file appears in the field.

Exporting text from a text field

You can export text from a text field using the Export command on the Tools menu.

➤ **To export text from a text field:**

1. Select the text field from which you want to export text.
2. From the Tools menu, point to Export, and then choose Field Text.

   The Export Text dialog box appears.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

3 In the Export Text dialog box, do the following:

- In the File name box, type a name for the text file.
- In the List files of type list, choose whether the file will be saved in ASCII format (with a .txt file extension) or in RTF (with an .rtf file extension).
- Under Directories, navigate to and select the directory where you want to export the file.

4 Click OK.

Instructor exports the text in a text file to the specified directory.

Importing text into record fields

When importing into record fields, you can import text files only in ASCII format. The ASCII file can be formatted as a fixed-field or delimited-field file.

Instructor automatically creates a new record field for each data field and a new page for each data record in the imported file. You cannot import data into a record field on the background of an existing page. Rather, when you import data into a new or existing book, Instructor creates a new page with a blank background and inserts new record fields as needed. All tabs and control characters are imported into Instructor along with the data.

The first new record field is always placed at the top of the background on layer 1, and each successive record field is placed below it. Instructor imports data from the first data field into the new record field on layer 1 of the first page and imports data from successive data fields into record fields on subsequent layers. Imported text is formatted according to the current settings in the Properties for Record Field dialog box. For more information about working with text and record fields, see Chapter 8, “Working with text.”

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index



Note Windows uses the ANSI (American National Standards Institute) character set for displaying characters. If you import data from text files into Instructor, you may need to retype some special characters. For details about special characters, see "Inserting special characters" in Chapter 8, "Working with text."

Importing fixed-field files

In fixed-field files, the number of characters in each data field of the source file determines where one data field ends and another begins. Whenever Instructor finds a record separator, such as a carriage return/linefeed (CRLF), it creates a new page and imports data from subsequent data fields into the record fields on that page until it finds another record separator. For example, most spreadsheet applications can save data as text in fixed-field files. Each cell, or column, in the spreadsheet is a data field with a fixed number of characters, and each row is a data record.

```
1/88    0,980.98    187.45    087.98    0,705.55¶
2/88    1,260.67    220.67    095.33    0,944.67¶
3/88    1,438.22    250.98    101.15    1,086.09¶
4/88    1,799.26    283.40    125.99    1,389.97¶
```

Figure 1
Table saved in fixed-field format

When you import a fixed-field file, you indicate how many data fields are in a data record by specifying the length of each field. Instructor uses the information about data field sizes as a template, so you must determine the precise length of each data field before you import the file.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To import a fixed-field file:**

Using the interface

1. Start a new book, or navigate to the page in an existing book after which you want to insert the pages with imported text.
2. From the Text menu, choose Character.

   The Character dialog box appears.
3. In the Character dialog box, select the font and point size for the imported text.
4. Click OK.
5. From the Insert menu, choose Pages.

   The Import dialog box appears.
6. In the Import type list, select Text files (*.txt).
7. Click Format to display the Text Format dialog box.
8. Choose Text—Fixed fields from the File format options.
9. In the Field lengths box, type the number of characters in each data field, separating them with commas. You can type any number of field lengths.

   For example, type *3,5,8,12* if the first data record in the source file contains four data fields with 3, 5, 8, and 12 characters, respectively. If the field lengths specified don't match those in the data file, the imported text won't be divided properly among record fields.
10. Click OK to return to the Import dialog box.
11. Navigate to and select the file you want to import, and then click OK.

Instructor imports the file, creating a new record field for each data field.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

- To import a fixed-field text file, use the *import* command with the *fixed* keyword and a list of field lengths:

```
--Imports a fixed-field text file that contains four fields
import "c:\data\customer.txt" as fixed using 3,5,8,12
```



For details about using the *import* command, refer to its entry in the OpenScript reference in online Help.

Importing delimited-field files

In delimited-field files, data fields can be any length and are separated by a delimiter. A delimiter is a single character, such as a comma or a tab, that separates data fields in a data record. Data records are separated by a CRLF. For example, a mailing list with names and addresses that vary in length is a delimited-field file.

```
John Q. Franklin,124 N.E. 8th Street,Bath,WA,98842,555-1029¶
Sam R. Jones,154 Ridge Road,Bath,WA,98799,412-7890¶
Sally F. Smith,23509 S.E. 91st Place,Lyme,WA,98679,302-5555¶
```

Figure 2
Mailing list in delimited-field format

Instructor creates a record field for each data field in the source file's first data record (the first line of the file). If any data record contains more data fields than the first data record, Instructor discards the extra data. If any data record contains fewer data fields than the first data record, Instructor leaves the remaining record fields blank on that page.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Instructor inserts the data into record fields of a fixed size, regardless of the length of the data fields. You can display data longer than the field by resizing the field on the background.

You can import a delimiter character as data by enclosing it in quotation marks (“ ”). For example, you might have a comma separating city and state names in the source file. If you want to keep that comma even though you’re using a comma as the delimiter between fields, you can import it as text by typing a quotation mark before and after the field containing the comma. The text in the source file should read *“Bellevue, WA”* instead of *“Bellevue”, “WA”*.

All field delimiters and record delimiters are ignored in quoted regions of text in imported files. To preserve quotation marks within a quoted region in the source file, enclose them within a second set of quotation marks. For example, the field in the source file must contain *“this is a ““quoted”” region”* to import *this is a “quoted” region* into Instructor.

➤ To import a delimited-field file:

Using the interface

1. Start a new book, or navigate to the page in an existing book after which you want to insert the pages with imported text.
2. From the Text menu, choose Character.

   The Character dialog box appears.
3. Select the font and point size for the imported text.
4. Click OK.
5. From the Insert menu, choose Pages.

   The Import dialog box appears.
6. In the Import type list, select Text files (*.txt).

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

7 Click Format to display the Text Format dialog box.

8 Choose Text—Delimited fields from the File format options, and then type a delimiter character in the Delimiter box.

- For a tab-delimited source file, type ^*t*.
- For a single-character delimiter, such as a comma or pound sign (#), type that character.

You cannot use a quotation mark (") as the delimiter. Instructor reserves the use of the quotation mark to enclose text that contains a delimiter in the source file. You can import a delimiter character as data by enclosing it in quotation marks.

9 Click OK to return to the Import dialog box.

10 Navigate to and select the text file you want to import, and then click OK.

Instructor imports the file, creating a new record field for each data field. Whenever Instructor finds a delimiter in the source file, it removes this character and places the data from the data field into a record field. Whenever Instructor finds a record separator (CRLF) in the source file, it creates a new page.

Using OpenScript

- To import a delimited-field text file, use the *import* command with the *delimited* keyword:

```
--Imports a delimited text file that uses spaces to separate fields
import "c:\titles.dbs" as delimited using space
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Exporting text from record fields

You can export data from Instructor record fields to text files so that you can use the data in other applications. You can export a record field of any size and export an unlimited number of data records from a book.

When exporting to both fixed-field and delimited-field file formats, Instructor exports data from all record fields on all pages that share the background of the page displayed when you choose the Export command. Instructor exports data from each page in the layer order of the record fields containing the data, starting with the record field on the lowest layer (see Figure 3).

In the text file, Instructor adds a carriage return/linefeed after each Instructor page. If you choose to export data in delimited-field format, Instructor also adds quotation marks around the contents exported from each record field.

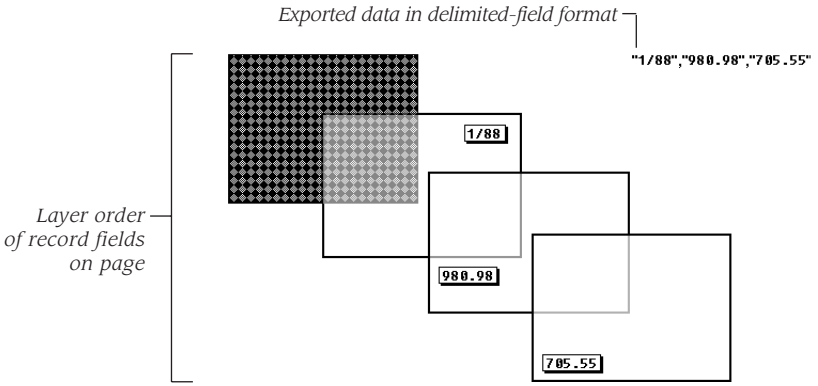


Figure 3
Data exported from a page

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Note Windows uses the ANSI character set for displaying characters. If you open an exported file with another application, you may need to retype some special characters. For details about special characters, see "Inserting special characters" in Chapter 8, "Working with text."

Exporting to fixed-field files

Figure 4 shows data from record fields exported to a fixed-field file. The first data record in the file contains data from the record fields on page 1, the next data record contains data from the record fields on page 2, and so on. In Figure 4, you'll notice that the first exported field in the first data record is truncated. When you export to fixed-field files, you must specify the number of characters in each field. If a field is larger than the specified size, text is truncated on the right side.

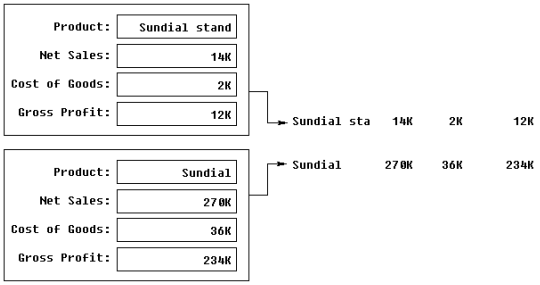


Figure 4
Exporting data to a fixed-field file

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ To export data to a fixed-field text file:

Using the interface

1. Open the book from which you want to export data.

   If the book uses more than one background, go to a page with the record fields containing the data you want to export.

2. From the Tools menu, point to Export, and then choose Record Fields.

   The Export dialog box appears.

3. In the Export dialog box, specify a path under Directories and type a file name in the File name box. If you don't type a file extension, Instructor adds the .txt extension.

4. Click Format to display the Text Format dialog box.

5. Choose Text—Fixed fields from the File format options.

6. In the Field lengths box, type the number of characters in each data field, separated by commas.

   For example, type *3,5,8,12* if you want the data records to have four data fields with 3, 5, 8, and 12 characters, respectively. You can type as many field lengths as needed to define the data fields.

7. Click OK to return to the Export dialog box, and then click OK again.

   Instructor exports the contents of the record fields to the file name you specified. The data in each record field becomes a data field in the text file. As Instructor exports the last record field on each page, it starts a new data record in the text file.



Note Instructor adds spaces to data fields if the field length you specify is greater than the number of characters in the original record field. Use positive numbers to export data aligned on the left, with fill space inserted on the right. Use negative numbers to export data aligned on the right, with fill spaces inserted on the left. If a field is larger than the specified size, text is truncated on the right side.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- To export text from record fields to a fixed-field text file, use the *export* command with the *fixed* keyword, specifying the length of the fields:

```
--Exports text from a book with 3 record fields
put "40,-80,10" into myFields          --2nd field is right-aligned
export "mybook.dat" as fixed using myFields
```



For more details about using the *export* command, refer to its entry in the OpenScript reference in online Help.

Exporting to delimited-field files

Figure 5 shows data from record fields exported to a delimited-field file. The first data record in the file contains data from the record fields on page 1, the next data record contains data from the record fields on page 2, and so on. The data fields are separated by the delimiter character you specify in the Export dialog box.

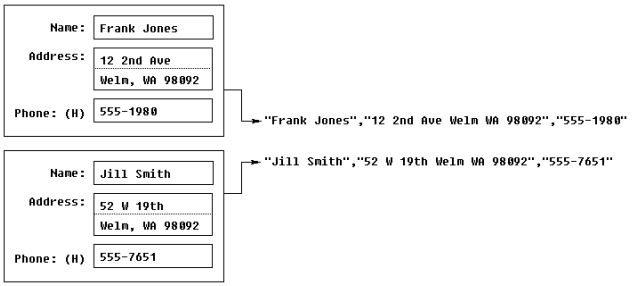


Figure 5
Exporting data to a delimited-field file

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ To export data into a delimited-field text file:

Using the interface

1. Open the book from which you want to export data.

   If the book uses more than one background, go to a page with the record fields containing the data you want to export.

2. From the Tools menu, point to Export, and then choose Record Fields.

   The Export dialog box appears.

3. In the Export dialog box, specify a path under Directories and type a file name in the File name box. If you don't type a file name extension, Instructor adds the .txt extension.

4. Click Format to open the Text Format dialog box.

5. Choose Text—Delimited fields from the File format options, and then type a delimiter character in the Delimiter box.

   - For a tab-delimited source file, type ^*t*.
   - For a single-character delimiter, such as a comma or pound sign (#), type that character.

   You cannot use a quotation mark (") as the delimiter. Instructor reserves the use of the quotation mark as a symbol to enclose text that contains a delimiter in the source file. You can export a delimiter character as data by enclosing it in quotation marks.

6. Click OK to return to the Export dialog box, and then click OK again.

   Instructor exports the contents of the record fields to the file name you specified. The data in each record field becomes a data field in the text file, enclosed in quotation marks.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- To export text from record fields to a delimited-field text file, use the *export* command with the *delimited* keyword and a character to use as the delimiter:

```
--Exports text to a text file, putting ! between each field
export "temp.dat" as delimited using "!"
```



For details about using the *export* command, refer to its entry in the OpenScript reference in online Help.

Importing and exporting graphics

You can import graphics from many file formats and can export graphics to the BMP, GIF, and JPEG formats. In general, you will want to import graphics into the Resource Manager dialog box so that you can use a graphic multiple times without increasing the file size of your application. For more information about using graphics as resources, see Chapter 9, “Using graphics,” and Chapter 17, “Using resources.”

However, on occasions when you plan to use a graphic only once, or when you want to place an image on your page quickly, you can import a graphic using the Graphic command on the Insert menu.

➤ **To import a graphic:**

1. From the Insert menu, choose Graphic.
2. In the Import Graphic dialog box, navigate to and select the graphic file you want to import, and then click OK.

   Instructor imports the graphic and places it on your page.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index



Note When you import a graphic, the newly created graphic object will either be a paint object or a picture object, depending on the file format of the original graphic. For more information, see Chapter 9, "Using graphics."

You can export a graphic from Instructor by selecting the graphic and choosing the Export command from the Tools menu. When you export a graphic, Instructor saves a copy of the image in the format you specify.

➤ **To export a graphic:**

1. Select the graphic you want to export.
2. From the Tools menu, point to Export, and then choose Graphic.
3. In the Export Graphic dialog box, do the following:
   - In the File name box, type a name for the graphic file.
   - In the List files of type list, select a file format in which the graphic will be saved.
   - Under Directories, navigate to the directory to which you want to export the graphic.
4. Click OK.

   Instructor exports the graphic to the specified directory.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Importing books and pages

You can import entire ToolBook II books or specific pages into a new or existing book. When you import an entire book, Instructor inserts all of the imported book's pages, backgrounds, viewers, and referenced resources into the current book. This method saves time when you want to use all of the information in a book without recreating it. Importing pages into a book is useful when you want to use the contents of a specific page.

When you import a book or pages into another book, the book properties of the imported book do not get applied to the new book's properties. Book scripts will not be assigned to the new book from the imported book, nor will settings such as page size and hyperlink color. Resources that are used on the imported pages will be added to the new book's resource system. Resources that are not used on the imported pages will not be added to the new book.

Importing books

When you import a book, Instructor inserts all of its pages and backgrounds after the current page of the current book, assigning each one a new ID number. If the imported pages and backgrounds had names in the source book, these stay the same.

Instructor moves the pages that originally followed the current page to the end of the book and assigns them new page numbers, but their ID numbers and names remain.

For example, if you navigate to page 1 of a book that has two pages on a single background and then import a book that also has two pages on a single background, the page numbers, page ID numbers, and background ID numbers will be updated as shown in Figure 6.



Note Changes that you make to the imported pages, backgrounds, viewers, and resources do not affect the original book.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

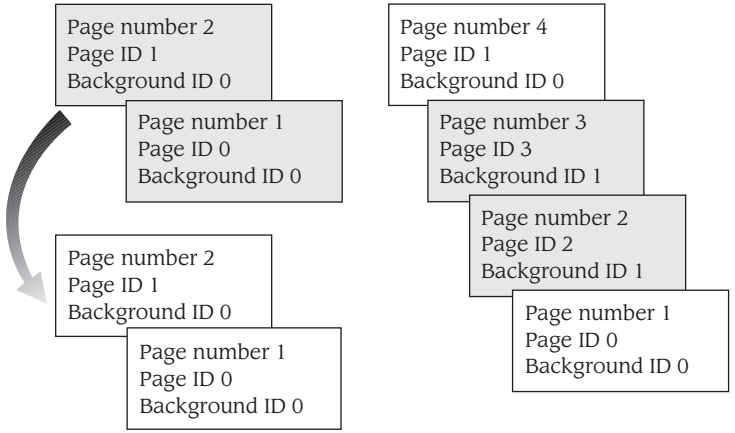
21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index



You can import a book into another book.

Instructor puts the pages and the background from the imported book after the currrent page.

Figure 6
Importing a book

➤ **To import a book:**

Using the interface

1 From a new book, or from the page in an existing book after which you want to insert the imported book, choose Pages from the Insert menu.

2 In the Import dialog box, select ToolBook files (*.?bk) in the Import type list.

3 Navigate to and select the book file you want to import in the File name box, and then click OK.

Instructor inserts all pages and backgrounds from the selected book into the current book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- To import a book, use the *import* command and specify a file name with the .tbk extension:

```
--Navigates to page 3
go to page 3
--Imports book after current page
import book "c:\mybooks\template.tbk"
```

Importing pages

You can import a page or range of pages from another book into a new or existing book. You can import the page with its background, or you can import the page onto the current background.

If you import a page or range of pages into an existing book, Instructor inserts the pages after the current page of the current book. Instructor changes the page numbers of the pages in the current book but does not change page ID numbers or background ID numbers. For details, see the preceding section, "Importing books."

If you import pages that contain record field text but don't import the corresponding backgrounds, Instructor follows these rules in sequence to determine where to place the record field text:

1. **Matching record field names** Instructor puts the imported text into a record field with a name that matches that of the original record field. For example, if you import a page containing text from a record field called *Student Name*, Instructor attempts to find a record field with the same name on the current background and place the text there. If no record fields match by name, Instructor proceeds to match by ID number.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

2 **Matching record field ID number** Instructor puts the imported text into a record field with an ID number that matches that of the original record field. If there is no match by either name or ID number, Instructor proceeds to match by layer order.

3 **Matching layers** Instructor puts the imported text in a record field that is on the same layer as the original record field. For example, the text from the record field at layer 3 on the original background is placed into the record field at layer 3 on the current background.

If the destination background contains more record fields than the source background, Instructor maps the record field's text as described above, leaving the extra record fields on the destination background empty. If the source background contains more record fields than the destination, Instructor maps the text as described above and discards the text of the extra record fields.

➤ To import a page or range of pages into your book:

Using the interface

1 From a new book, or from the page in an existing book after which you want to insert the imported pages, choose Pages from the Insert menu.

2 In the Import dialog box, select ToolBook files (*.?bk) in the Import type list.

3 Click Format to display the Book Format dialog box.

4 Choose Pages to indicate that you are importing a page range, and then enter the starting and ending pages to import in the From and To boxes, respectively. (To import a single page, enter its number as the starting and ending page.)

5 If you want to import the background along with the pages, select Include background.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

6 Click OK to return to the Import dialog box.

7 Navigate to and select the book file you want to import, and then click OK.

Instructor inserts the pages from the selected book into the current book. If you selected Include background, Instructor also imports the backgrounds of the pages you indicated.

Using OpenScript

- To import pages from a ToolBook II book, use the *import* command, specify a file name with the .tbk extension, and then indicate the page range to import:

```
--Navigates to end of book, then imports 15 pages from another book
--using the background of the current page
go to last page
import pages 2 to 16 of book "c:\mybooks\glossary" \
   without background
```

Customizing the import process

If you want to import only certain fields or records, or assign the imported text to existing pages or new pages on an existing background, you can create a script that customizes how Instructor imports the text data.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

The OpenScript *import* command always creates new pages on a new background. To assign text to an existing page or background, use the *openFile* and *readFile* commands. To control how much text is placed in each field, use the *readFile* command. For example:

```
--Imports a file that contains fixed-length records of 80 characters
to handle buttonClick
   DataFile = "c:\import.txt"           --File name
   clear sysError
   openFile DataFile
   readFile DataFile for 80
   while It is not null and sysError is null
      --Sends a message to another handler to process the record
      send processRecord It
      readFile DataFile for 80
   end while
   closeFile DataFile
end buttonClick
```

```
--Imports a text file into the text of a field and replaces
--CRLF characters with spaces
to handle buttonClick
   --set the filename in a variable to make the code more efficient
   --and to make it easier to change later
   dataFile = "C:\import.txt"
   openFile dataFile
   clear text of field "WP text"
   while sysError is not "end of file"
      readFile dataFile to LF
      if last char of It is CR
         clear last char of It          --Clears trailing CR character
      end if
      put space & It after text of field "WP text"
   end while
   closeFile dataFile
end buttonClick
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Chapter 19

Creating windows with viewers

Viewers are windows that display pages. The ToolBook II Instructor main window is a viewer, as are pop-up windows, tool palettes, dialog boxes, startup screens, and other visual tools that enhance both your applications and your authoring environment. This chapter describes how to create and display viewers, and shows you how to use viewers in your application.

IN THIS CHAPTER

About viewers **494**

Creating viewers **503**

Setting viewer properties **508**

Working with viewer properties **510**

Using viewers in an application **520**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

About viewers

Viewers are windows that you create in ToolBook II Instructor to display pages. The Instructor main window is the default viewer, but you can create additional viewers to show multiple pages at once. You can use viewers to display:

- windows that show other pages from the application.
- dialog boxes.
- floating palettes.
- toolbars.
- status bars.
- pop-up graphics or text boxes.
- logos and application startup screens.

Viewers, like any other object in Instructor, are defined by properties that determine their appearance and behavior. Working with viewers requires some experience with Instructor and the OpenScript programming language. To help you get started with viewers, this section provides an introduction to using viewers.

Using viewers on the Internet

You can use all types of viewers in a native ToolBook II application (TBK file) distributed over the Internet using the ToolBook II Neuron browser plug-in. If you plan to export your book to DHTML, do not create viewers with the New Viewer menu command; they will not export to DHTML.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

A viewer's default page

When you create a viewer, you assign it (or allow Instructor to assign it) a default page that appears when the viewer is shown. Some viewers may display only that default page—a simple dialog box, for example. You can also display and allow navigation to other pages in the viewer, just as you can in the main window. For example, if you are showing pages from a tutorial in a viewer, the default page might contain controls allowing the user to move back and forth in the lesson. The default page that you assign to a viewer can be a page from the current book or one from another book.

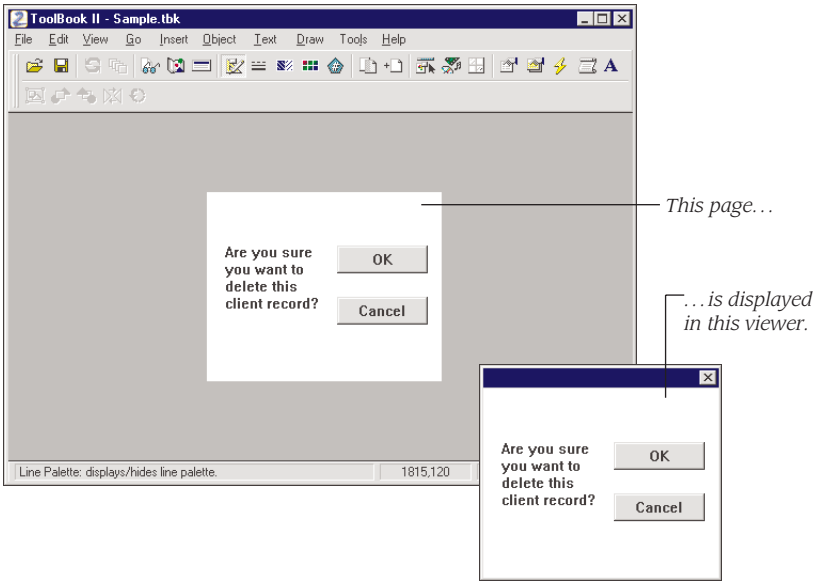


Figure 1
Pages are displayed in viewers

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Tip To print a viewer's pages, display the book that contains the viewer's pages and then print the pages. You cannot print pages directly from the viewer.

Types of viewers

You can create viewers in a variety of styles, depending on the purpose of the viewer. Instructor has six viewer templates available, each with certain visual characteristics and preprogrammed behavior. For example, a viewer for a dialog box has a special border and no Maximize or Minimize buttons, so it can be moved but not resized.

Whether you use a viewer template or create a custom viewer, you can set properties for viewers to refine their appearance and functionality. Border style, position, color, and hide-and-show behavior are among the many viewer properties you can set.

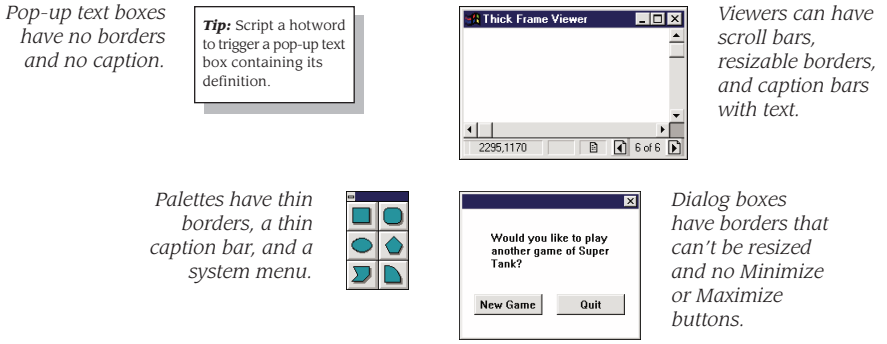


Figure 2
Viewers exhibit different styles and behavior

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Viewer behavior

A viewer can appear as a pop-up window, which can float anywhere on the screen, or as a child window, which appears inside another window. Pop-up viewers are useful for displaying dialog boxes, palettes, and Help text. Toolbars and status bars are best displayed as child viewers.

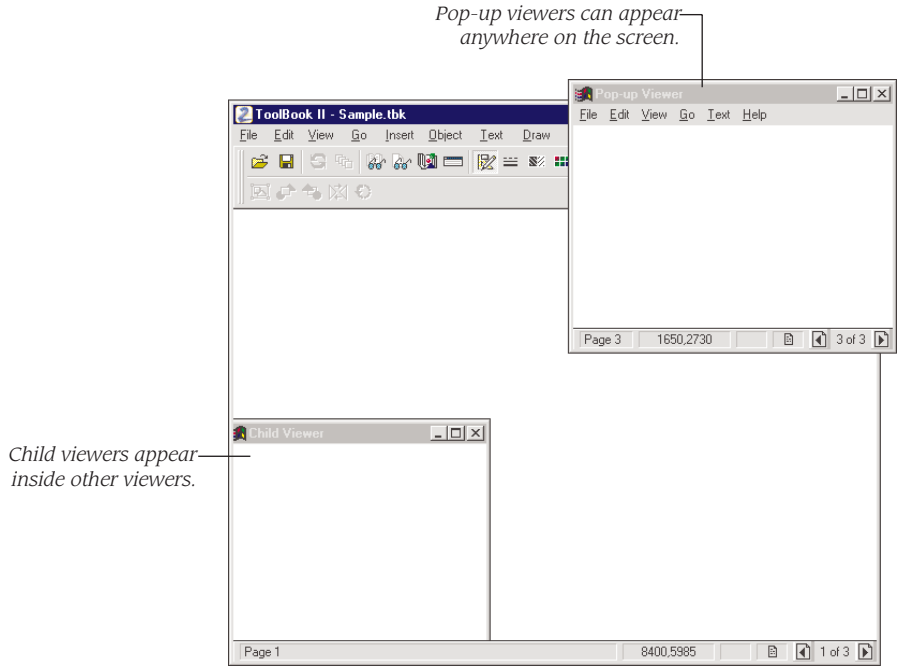


Figure 3
Pop-up viewers and child viewers

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

A viewer, whether it is a pop-up viewer or a child viewer, is always owned by another window, called the parent window. The parent window determines the viewer's behavior. For example, when you minimize the Instructor main window, all the Instructor palettes are minimized as well, because the main window is their parent window. You can set any window to be the parent of a viewer, including windows that belong to other Windows programs. By changing a viewer's parent window, you can create applications that are independent of the Instructor main window, such as a floating calculator or calendar.

A child viewer differs from a pop-up viewer because it appears only inside the parent window and is clipped—or obscured—if a user moves an edge outside the edge of the parent window. Child windows can also be tiled—or snapped—against an edge of their parent window. When a viewer is tiled, its dimensions automatically increase or decrease when its parent window is resized.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

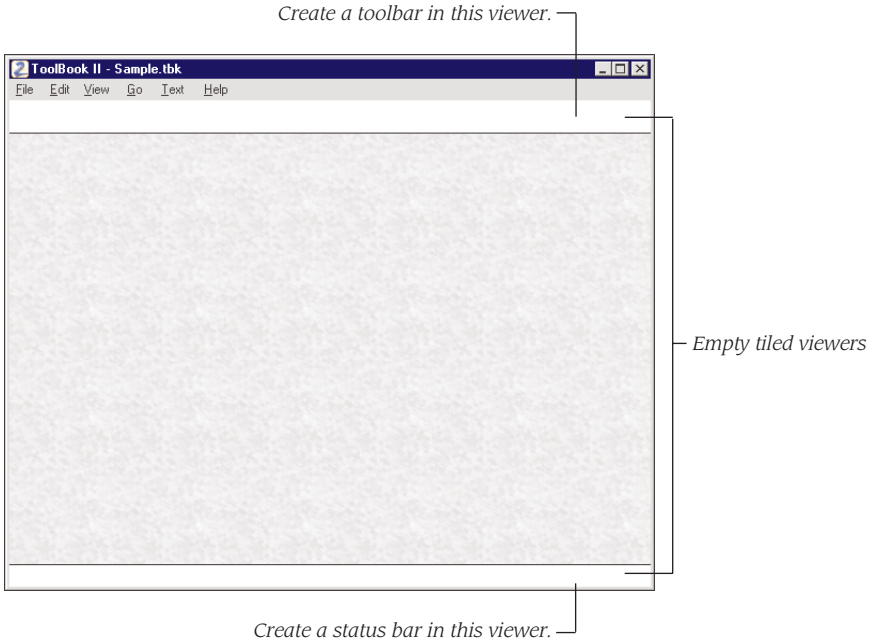


Figure 4
A tiled viewer adheres to one side of the parent window

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Differences between pop-up viewers and child viewers

Pop-up viewers	Child viewers
Will not work when exported to DHTML	If set to be tiled, will be converted to frames when exported to an Internet-based file format
Float on top of parent windows	Are clipped by parent windows
Can have menus	Cannot have menus
Have a highlighted title bar when activated	Do not have a highlighted title bar when activated
Appear as icons on the Windows desktop when minimized	Appear as icons inside parent window when minimized

Elements of a viewer

Each viewer, including the Instructor main window, is made up of a frame window and a client window (see Figure 5). In most cases, the client window takes up the entire client area. However, if the client window is smaller than the client area, such as when the viewer is maximized, the viewer's mat is visible. If you tile a viewer, its client window is attached to the frame of its parent window, and the parent window's client window is moved to accommodate the tiled window.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index



Figure 5
A viewer consists of a frame window and a client window

Understanding viewers and the object hierarchy

Before you incorporate viewers into an application, you should understand how they fit into the object hierarchy. A ToolBook II object forwards messages—communications that an event, like a button click, has occurred—to its parent object. The parent of an object (such as a button) on a page is either a group (if the button is grouped) or the page. The page's parent is the background, the background's parent is the book, and the book's parent is the Instructor system.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Viewers have their own branch of the object hierarchy. They are not the parents of the pages they display, so they do not receive messages forwarded from the page. However, viewers receive their own messages, such as the enter and leave event messages *openWindow* and *closeWindow,* and the notification messages *sized*, *moved*, *hidden*, and *shown*. The parent object for viewers is the book, which receives any messages the viewer forwards or does not handle.



Note A parent object is different from a parent window. A viewer's parent object is the object above it in the object hierarchy (the book) and to which the viewer will forward messages. A viewer's parent window is the window that controls how the viewer behaves, such as when the viewer will be minimized.

If different viewers are displaying pages from different books, Instructor sends messages up different object hierarchies. For example, the main window might display a page from one book, while a dialog box viewer displays a page from another book. If the user clicks a button in the dialog box viewer, the message travels up the object hierarchy of the viewer's page. Any *buttonClick* handlers in the script of the book displaying the main window will not receive the message; therefore, clicking in the main window and in the tool palette might result in different behavior.



For details about the object hierarchy, refer to the *Programming in OpenScript* online book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Creating viewers

You can create a viewer in two ways. You can:

- select a viewer template, which includes properties that are preprogrammed for one of six common uses.
- create a custom viewer, and then set its properties.

Using a viewer template saves time: Templates come equipped with features you would otherwise have to set manually, one by one. You can further customize viewers created from a template by setting or refining individual properties. However, if you have to customize a viewer template extensively, it may be faster to create a custom viewer. Instructor includes several viewer templates, as shown in Figure 6.

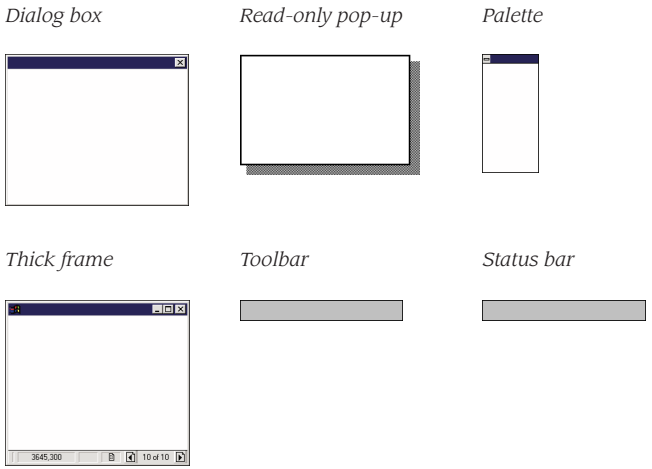


Figure 6
Viewer templates

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ To create a viewer using a viewer template:

- From the Insert menu, point to New Viewer, and then choose a viewer template option from the submenu.

  Instructor creates a new viewer based on the template you choose, creates a new page and background as its default page, and then displays the new viewer.



Note When you create a viewer using a viewer template, Instructor automatically adds a new page and background to your book that are assigned as the viewer's default page. This page is set to be skipped when a user navigates through the book.

➤ To create a custom viewer:

Using the interface

1. From the Object menu, choose Viewers.
2. In the Viewers dialog box, click New.

   The New Viewer dialog box appears.
3. Under Initial property settings, choose Custom.

   (Choosing Use template allows you to select a viewer from the template options.)
4. Select Create new page and background to create a new page and background for the viewer, and then click OK. (If you do not select Create new page and background, Instructor uses the page and background currently displayed in the main window as the default page.)

   Instructor creates a new viewer and displays the Properties for Viewer dialog box. You can modify the viewer's properties now, or click Done in the Viewers dialog box.
5. *Optional.* Close the Properties for Viewer dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

- To create a new viewer, use the *new viewer* command. Instructor creates a new pop-up viewer with a thick frame, assigns it the next available viewer ID number, and assigns the page currently displayed in the main window as the viewer's default page. You can change the viewer's properties after Instructor has created it. When Instructor creates the viewer, it places the viewer's *uniqueName* into *It*. For example:

```
new viewer
--Assigns new viewer's object reference to newViewerID
newViewerID = It
--Assigns name to viewer
name of newViewerID = "myPalette"
--Creates child viewer
defaultType of viewer "myPalette" = child
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Restricting navigation

Let's say you create a page in your book that you want to display in a tool palette viewer, but you don't want users to encounter that page while navigating through other pages in the book. Using either the Properties for Page dialog box or OpenScript, you can prevent users from navigating to a page intended for display in a viewer.

➤ **To prevent navigation to a page at Reader level:**

Using the interface

1. Navigate to the page displayed in the viewer.
2. From the Object menu, choose Properties for Page.
3. On the Behavior tab, under Navigation, select Skip this page when navigating.
4. *Optional.* Close the Properties for Page dialog box.

Using OpenScript

- By setting the page's *skipNavigation* property to *true,* you cause Instructor to skip over that page during navigation at Reader level when menu-event messages such as *next*, *previous*, *last*, *first*, and *flip (*or the *go* command with these values) are sent. However, you can still navigate to the page at Author level or by using an explicit reference to the page name or ID number:

```
go to page "dialog"
go to page 2
currentPage of viewer ID 2 = "dialog"
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Copying and pasting viewers

To save time, you can copy a viewer that you've created previously. The copied viewer can be pasted into the current book or into a different book. You copy a viewer using the Copy and Paste buttons in the Viewers dialog box.

➤ **To copy and paste a viewer:**

1. From the Object menu, choose Viewers.

   The Viewers dialog box appears.

2. Select the viewer you want to copy in the list, and then click Copy.

   The viewer is copied to the Clipboard.

3. Do one of the following:

   - To paste the viewer into the current book, click Paste.
   - To paste the viewer into another book, open the book into which you want to paste, choose Viewers from the Object menu, and then click Paste in the Viewers dialog box.

4. Click Show Viewer to display the viewer.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Setting viewer properties

You set a viewer's properties to define its appearance and behavior, such as the title in a viewer's caption, its default position and size, and the icon it displays when minimized. You can also change the properties of any existing viewer, even if you created it with a viewer template.

Viewers support both persistent and nonpersistent properties. Persistent properties are saved when you close a viewer, remain with a viewer if you copy it into another book, and apply to the viewer each time it is opened. Viewer type, default page, position, and size are some examples of persistent properties.

Nonpersistent properties, such as a viewer's position and size at a particular time, apply when the viewer is displayed and can change as the user resizes, moves, minimizes, maximizes, or changes the viewer in other ways. Instructor discards nonpersistent properties when the viewer is closed. For example, you can change a viewer's nonpersistent position, but if you close the viewer and open it again, it will appear at its default position.

Some persistent viewer properties override the values of others. For example, if you create a viewer with Maximize and Minimize buttons but then change its border style to shadowed, the Maximize and Minimize buttons disappear.



For a complete listing of all persistent and nonpersistent viewer properties, refer to the OpenScript reference in online Help.

Setting a viewer's persistent properties

You can set a viewer's persistent properties using the Properties for Viewer dialog box or using OpenScript commands. To set a viewer's nonpersistent properties, you can move or resize the viewer when it is displayed to change its position or size, or use OpenScript to change other properties.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

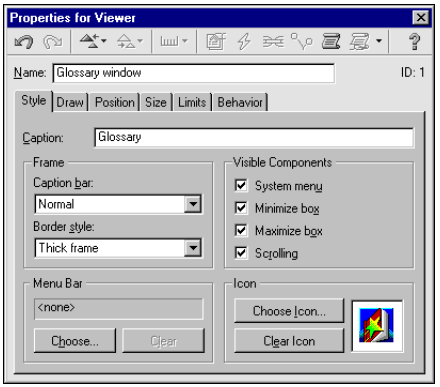


Figure 7
Properties for Viewer dialog box

➤ **To set a viewer's persistent properties:**

Using the interface

1 From the Object menu, choose Viewers.

The Viewers dialog box appears.

2 Select a viewer from the list, and then click Properties.

Instructor displays the Properties for Viewer dialog box.

3 Specify the property settings.

For details about specific viewer properties, see the following section, "Working with viewer properties."

4 *Optional*. Close the Properties for Viewer dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using OpenScript

- Set viewer properties using the name of the property and a viewer reference:

```
--Sets a viewer's name, default page, and default type
name of viewer ID 1 = "Map"
defaultPage of viewer "Map" = page 2
defaultType of viewer "Map" = popup
```



Tip For a shortcut to the Properties for Viewer dialog box, right-click the viewer's caption bar or frame and click the Properties button on the right-click menu toolbar. Not all viewers have a caption bar, and the frame can be difficult to select. Keep in mind that the main window is a viewer. If you click the main window while the Properties for Viewer dialog box is open, the viewer properties for the main window will appear.

Working with viewer properties

You can set a viewer's properties to define its appearance and behavior, such as the caption bar's style and text, the viewer's border style, and whether the viewer displays a status bar. These properties determine how the viewer looks, as well as how the user can work with the viewer.

You can also assign menus and icons to viewers by specifying properties. Menus allow users to select viewer-specific options. If you assign an icon to a viewer, for example, the icon is displayed when the viewer is minimized.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Setting caption style

Viewers can have a normal or thin caption bar, or no caption bar at all. If you choose to include a caption bar, the viewer can be moved by dragging and the caption bar can display text. In addition, a caption bar can contain a system menu as well as Maximize, Minimize, and Close buttons. Setting a viewer's border style to shadowed automatically disables the caption bar, as does tiling a child viewer.

No caption bar

Caption bar with text

Thin caption bar without text

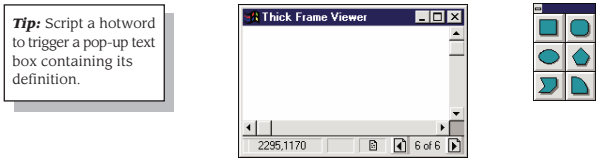


Figure 8
Viewer caption styles

Assigning a menu bar to a pop-up viewer

You can add a menu to a pop-up viewer by assigning it a menu bar resource. When a pop-up viewer displays a menu bar, menu-event messages are sent to the page displayed in the viewer, and *enterMenu* and *menuItemSelected* messages are sent to the viewer. Menu bars appear only if the border style of the viewer is set to none, thin frame, or thick frame.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Assigning an icon to a viewer

You can assign an icon resource to any viewer for Instructor to display when the viewer is minimized. The icon for a child viewer is displayed in its parent window. The icon also appears in the caption bar and, when the window is minimized, appears in the Windows taskbar.

For details about creating, importing, and working with icons, see Chapter 17, “Using resources.”

Setting a viewer’s mat color

You can set the color of a viewer’s mat to a custom color or use the color specified in the Windows Control Panel. All other colors for a viewer, including its border and caption, are determined by the corresponding color settings in the Windows Control Panel.

Using image buffers to optimize viewer display

You can control how fast, efficiently, and smoothly Instructor draws pages in viewers by setting a viewer’s *imageBuffers* OpenScript property. An image buffer is an area in your computer’s memory in which Instructor draws objects. When all the objects have been drawn, Instructor displays them all at once by copying the image buffer to the screen.

The benefit of an image buffer is higher display quality—objects appear and move smoothly, with no flickering. For example, an image buffer can make animation sequences look better. Using image buffers will also improve imaging speed when navigating between pages. However, because Instructor has to draw the objects and then transfer them to the screen, using image buffers can result in a slower page display. In addition, image buffers require memory. Therefore, use image buffers only when you require the higher display quality that they provide.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

You can assign zero, one, or two image buffers to a viewer. Each viewer template has a unique default. If you assign one buffer, Instructor uses it to draw the page and the background as needed. One image buffer is recommended for viewers that display animation sequences, but where no navigation between pages will take place.

If you assign two image buffers, Instructor uses one to draw pages and the other to draw backgrounds, which is especially useful if you need quick navigation between pages that share the same complex background, since Instructor does not have to redraw the background for each page. Using two image buffers will also enhance imaging performance if you move, show, or hide objects on a page on top of a complex background.

If the page contains objects that do not move, use zero image buffers to save memory and improve drawing speed. For example, assign zero image buffers to a viewer that displays a dialog box.



Note If you assign zero image buffers to a viewer, all objects on the pages and backgrounds it displays are drawn directly on the screen, even if their *drawDirect* property is set to *false*.

If memory is low on your computer when you display a viewer, Instructor might not be able to create the number of image buffers you have assigned to the viewer. In that case, Instructor tries to display the viewer with a lower number of image buffers, repeating the process if necessary. If Instructor reduces the number of image buffers automatically, it affects only that instance of the viewer and has no permanent effect on the number of image buffers you have assigned.

After you have established the number of image buffers for a viewer, you can change the setting at any time. If you change the number of image buffers for a viewer already displayed, Instructor redraws the viewer and its contents to reflect the change.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Adding a status bar to a viewer

Viewers can display status bars that look like the status bar in the main window. You can specify that a status bar appear by default at Author level, Reader level, or both. If the status bar is visible, Instructor displays Help text for the menu bar assigned to a viewer, and you can display custom text such as progress messages.

The OpenScript properties *authorStatusBar* and *readerStatusBar* specify whether a status bar appears by default at Author level and Reader level, respectively. If these properties are *true,* the status bar appears; if they are *false,* the status bar does not appear. However, these properties specify only the default appearance of the status bar. Regardless of the values of these properties, you can show or hide the status bar by pressing F12 or using OpenScript commands.

Setting viewer position

You can set a viewer's default position and tiling behavior by choosing viewer position options. The default position determines where the viewer appears when you first show it. Tiling behavior refers to the way a child viewer (a toolbar or status bar, for example) appears in its parent viewer's client window (see Figure 9). Toolbars and custom status bars are usually tiled. You can also specify that a viewer appear on top of other viewers, which is useful for viewers displaying palettes and logos.

Instructor stores viewer coordinates in pixels, even though you can establish values in page units in the Properties for Viewer dialog box. For details about converting between these two units of measurement, see "Positioning viewers using coordinates," later in this chapter.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Setting viewer size

By default, the OpenScript *autoSize* property of most new viewers is *true*, which causes Instructor to automatically resize the viewer's client area to the background of the page it displays. (The *autoSize* property of a new viewer is *false* if you use the toolbar and status bar templates.) However, you can establish a custom default size for a viewer's client window instead, so that the viewer stays the same size no matter what page it displays.

If the viewer has a thick frame, users can resize the viewer while it is open, but it reverts to its default size when closed. You can also establish the minimum and maximum size to which viewers with thick frames can be resized.

If the viewer's size is larger than the page it displays, the viewer's mat becomes visible. By default, the page is centered in the client area, but you can position it instead in the upper-left corner of the client area.

You can specify that the viewer initially appear in its maximized or minimized state. You can also lock a viewer in its minimized state, which is useful if you want the viewer to be available but out of the way.

Positioning viewers using coordinates

By default, Instructor measures the position of viewers in pixels. Pixel size is dependent upon the resolution and size of your monitor. If a window is positioned at a point located 100 pixels from the top and left sides of the screen (*100,100*), its actual location on the screen would depend on the screen resolution. At a resolution of 640 by 480 pixels, the window is farther from the upper-left corner of the screen than at a resolution of 1024 by 768 pixels.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

By contrast, Instructor objects such as pages, buttons, and fields, as well as the position of the mouse, are measured in page units. Page units measure the space between an object and the upper-left corner of the window. There can be different numbers of page units per pixel depending on the video device driver and the Windows settings of your user.

Although Instructor measures viewer position in pixels, it measures the viewer properties *defaultClientSize* and *clientSize* in page units. The size of a viewer is determined by the size of its client area. By measuring viewer size in page units, Instructor can ensure that the viewer size remains the same on different display devices relative to objects on the page.

You can change the units of measurement you work in as you position or size a viewer or object by clicking the Set Units button on the toolbar of the Properties dialog box. You may want to start out using the default settings and adapt them later on, if the need arises. Neither pixels nor page units guarantee the physical size of an object, which depends on the combination of system settings and hardware size. There are three things that affect the physical size of an object:

- The resolution of the screen (for example, 640 by 480 pixels)
- The number of page units per pixel (specified as a Windows setting)
- The size of the monitor

Instructor provides functions that you can use to convert coordinate units between pixels and page units, which is useful if you want to align viewers with objects on a page. For example, if you want a viewer to appear exactly where the user clicked the mouse, you would determine the mouse position in page units, convert those coordinates to pixels, and then use the new units to set the viewer's position.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

To determine the position of viewers, Instructor uses one of three reference points in pixels: screen, frame, or client. You position a viewer in relation to the *0,0* position of one of these reference points. To position an object in page units, Instructor uses the upper-left corner of the viewer's client area as the *0,0* point.

To set the position of a viewer, you have to know where the *0,0* point is for that type of viewer. For example, if you execute the statement *set position of viewer ID 3 to 0,0*, the exact location to which the viewer moves depends on the type of viewer. The following table specifies the reference points for different types of viewers.

Reference points for viewer coordinates

Viewer type	Reference point for coordinates
Pop-up window, main window	Screen
Untiled child	Client window of parent*
Tiled child	Frame of parent

* This is the default option; if you set the *parentHandle* of the child viewer to the window handle of its parent, it uses its parent's frame for reference.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

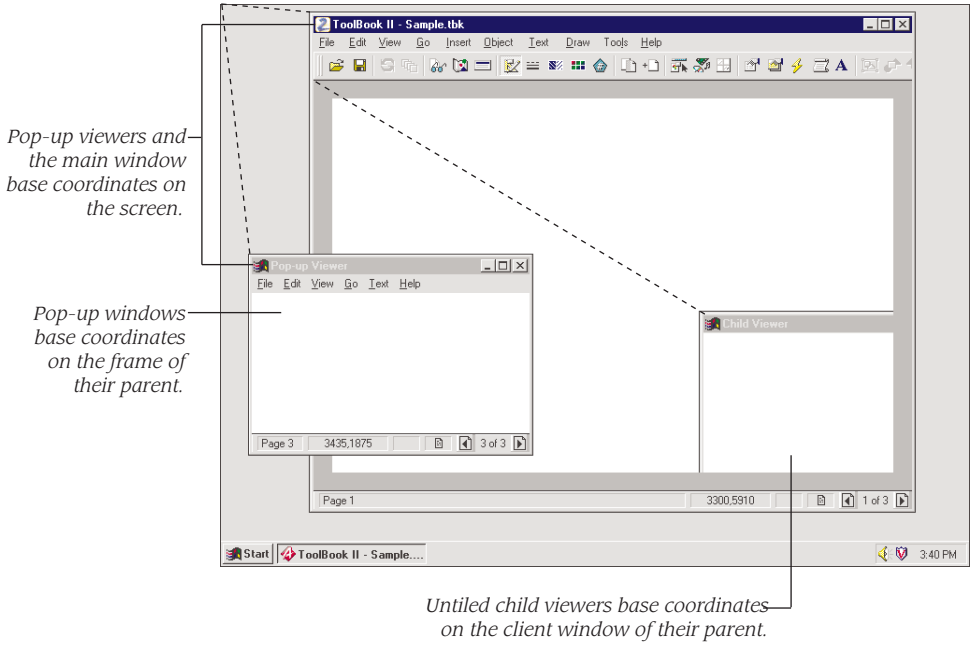


Figure 9
Viewers use different reference points for position

Instructor provides the functions shown in the following table so that you can convert a set of coordinates based on one reference point into those based on another.

Note To determine the number of page units per pixel for the current display device, use the system property *sysPageUnitsPerPixel*. For details, refer to its entry in the OpenScript reference in online Help.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Coordinate conversion functions

Use this function...	To do this...
pageUnitsToScreen( )	Align a pop-up window with an object on a page
screenToPageUnits( )	Align an object on the page with a pop-up window
pageUnitsToFrame( )	Align a tiled window with an object on the page
frameToPageUnits( )	Align an object on the page with a tiled window
pageUnitsToClient( )	Align an untiled child window with an object on the page
clientToPageUnits( )	Align an object on the page with an untiled child window
clientToScreen( )	Align a pop-up window with an untiled child window
screenToClient( )	Align an untiled child window with a pop-up window
frameToScreen( )	Align a pop-up window with a tiled window
screenToFrame( )	Align a tiled window with a pop-up window

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Following is an example of using the location where the user clicks on a hotword as the location to display a pop-up text box containing the hotword's definition:

```
to handle buttonClick loc
   if object of target = "hotword" then
      pageName = text of target
      definitionViewer = viewer "definition"
      if isOpen of definitonViewer = false then
         defaultPage of definitionViewer = page pageName
         open definitionViewer
         clear defaultPage of definitionViewer
      else
         currentPage of definitionViewer = page pageName
      end if
      pos = pageUnitsToScreen (loc targetWindow)
      position of definitionViewer = pos
      show definitionViewer
   end if
end
```

Using viewers in an application

After you have created a viewer, you can use it in your application to display information. Working with viewers is slightly more complex than working with objects on a page, because you must manipulate the viewer as well as the page that it displays.

Unlike other objects, viewers do not remain displayed at all times; you must determine and specify the conditions of their display. Displaying a viewer involves more steps than hiding and showing other Instructor objects, because Instructor makes a distinction between opening and showing a viewer, and between hiding and closing it.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

To use a viewer while your application is running, you perform up to four steps, as listed below:

- **Open the viewer** Creates an instance of the viewer, but does not display it. When a viewer is open, you can set its nonpersistent properties, such as the name of a page to show.
- **Show the viewer** Displays the viewer and its current page. If you show a viewer without opening it first, Instructor opens the viewer automatically but will not have the opportunity to set nonpersistent properties before the viewer appears.
- **Hide the viewer** Makes the viewer disappear but does not close it. When a viewer is hidden, its nonpersistent properties are retained, and you can get or set them.
- **Close the viewer** Closes the viewer, frees the memory that it uses, and discards its nonpersistent properties. If you want to display the viewer again after closing it, you must reopen it.

To open, show, hide, and close viewers, use the OpenScript commands *open*, *show*, *hide*, and *close*. For example, you can open and display a pop-up text box in the *buttonClick* handler for a hotword, or show a viewer for a logo display during an *enterApplication* handler:

```
to handle enterApplication
   show viewer "Logo"
   --Further initialization statements here
   close viewer "Logo"
   forward
end
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

You can determine whether a viewer is open by using its *isOpen* property, and you can get a list of all open viewers by examining the value of the *sysOpenWindows* property. For example, the following *buttonClick* handler toggles a tool palette's visibility:

```
to handle buttonClick
   if isOpen of viewer "toolPalette" is true then
      close viewer "toolPalette"
   else
      show viewer "toolPalette"
   end if
end buttonClick
```

Certain actions you perform automatically close viewers, as do certain OpenScript commands. Instructor closes all viewers:

- that belong to a parent viewer when the parent is closed.
- owned by that book when a book is closed.
- when Instructor is closed.

Using color palettes with viewers

If you are working with 256-color display devices, you will want to use color palettes to avoid a flickering effect known as palette flash. When you activate a viewer, the page displayed in that viewer controls how colors are displayed. For example, if the page displayed in the viewer contains a paint object that has its own palette, Instructor redisplays the screen in the colors for that palette. Similarly, if the active viewer displays a page from another book, Instructor uses the book palette from that book to display all objects on the screen.



Note Because you cannot activate a child window, only pop-up windows control the color palette.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

If the palette is very different from the one used for pages in other viewers, the pages in nonactivated viewers might appear in unattractive colors, and you might experience palette flash. To avoid these problems, you can assign the same palette to all the books from which you display pages.

For details about applying palettes, see Chapter 9, "Using graphics."

Displaying modal and nonmodal viewers

Viewers are either modal or nonmodal. A modal viewer remains active until it is hidden or closed; the user cannot interact with the rest of your application while a modal viewer is shown. For example, dialog boxes are often modal. A nonmodal viewer can remain open while a user interacts with the rest of your application. For example, tool palettes are nonmodal; a user can leave them open while interacting with the rest of your application.

To display a modal viewer, use the *modal* keyword with the *show* command:

```
show viewer "dialog" as modal
```

Showing and hiding viewers automatically

By setting persistent viewer properties, you can control the viewer's display automatically. You can:

- open and show the viewer automatically at the time the book is opened, which is useful for displaying logos, toolbars, and status bars.
- hide the viewer when another Windows program becomes active, which is useful for displaying palettes.
- close the viewer when the user next clicks the mouse, which is useful for the pop-up text boxes commonly used in hypertext and online Help applications.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

- force the viewer to remain at Reader level even when the main window is at Author level, which is useful for viewers displaying tool palettes.
- hide the viewer at Reader level, which is useful for viewers displaying authoring tools.

➤ To set viewers to show and hide automatically:

Using the interface

1. From the Object menu, choose Viewers.

   The Viewers dialog box appears.

2. Select a viewer from the list, and then click Properties.

   Instructor displays the Properties for Viewer dialog box.

3. On the Behavior tab, select options to specify how you want the viewer to hide and show.

4. *Optional.* Close the Properties for Viewer dialog box.

Using OpenScript

- To set automatic show and hide behavior for a viewer, set these properties: *autoShow*, *hideOnDeactivate*, *alwaysReader*, *hideOnReader*, *autoClose*, and *alwaysOnTop*. All these properties take values of *true* or *false*. For example:

```
autoClose of viewer "definitions" = true
alwaysReader of viewer "myPalette" = true
```

Using OpenScript to display pages in viewers

When using OpenScript to display a page in a viewer, you generally create a script in a page, background, or book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

- To display a page in a viewer, you set its *currentPage* property:

```
--Displays a page in a viewer to show the definition of a hotword.
--The page must contain a field with a definition.
to handle buttonClick
   if object of target is "hotword" then
      --Opens the viewer
      pageName = text of target
      defViewer = viewer "definition"
      if isOpen of defViewer = false then
         defaultPage of defViewer = page pageName
         open defViewer
         clear defaultPage of defViewer
      else
         currentPage of defViewer = page pageName
      end if
      show defViewer
   end if
end buttonClick
```

- You can also navigate in a viewer using the *go* command, just as you would in the main window:

```
to handle buttonClick
   --Determines which viewer is affected
   in viewer "clipart"
      go to next page
   end in
end buttonClick
```



Note The *sysHistory* property applies only to the Instructor main window; when you navigate in viewers, you cannot use the History command from the Go menu or send the *back* message to return to a page previously displayed.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Saving changes to pages displayed in viewers

If you use viewers to display pages from more than one book, Instructor does not save changes to all the pages when the user chooses Save from the File menu. The Save command saves changes only to the book displayed in the main window. You can save changes only to pages from other books by using the OpenScript *save changes* command with a book reference:

```
save changes to book "clipart"
```

To save changes automatically, set a book's *saveOnClose* property to *yes,* which causes Instructor to save the book when a viewer displaying pages from that book is closed. You can also set the *saveOnClose* property not to save changes, or to prompt the user to save the book, depending on the value of a second property, *sysChangesDB*.



For details, refer to the entries for *saveOnClose* and *sysChangesDB* in the OpenScript reference in online Help.

Using *lockScreen* and *sysLockScreen* with viewers

Each viewer has a *lockScreen* property, which, if *true*, prevents Instructor from changing the screen image of that viewer until the current handler and any calling handlers are finished. For example, if you are showing an animation sequence and want to place objects at particular positions before starting, set the viewer's *lockScreen* property in an *enterPage* handler and then move the objects as necessary. The next time Instructor is idle, or when you set *lockScreen* to *false*, Instructor updates the screen image with your changes. The *lockScreen* property affects any change to the viewer, including hiding, showing, minimizing, maximizing, or restoring the viewer.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

You can also set the *sysLockScreen* property, which applies to all viewers and overrides the effect of a viewer's *lockScreen* property. You could set this property before making changes to several viewers. If you show a viewer as modal while *sysLockScreen* is *true*, Instructor sets *sysLockScreen* to *false* in order to prevent a conflict between the state of the property and the modal nature of the viewer.

Working with the target window and parent window

If more than one viewer appears onscreen, Instructor uses the target window to determine which viewer contains the page that is the context for OpenScript commands and object references. The target window is the window that contains the object that received the original message.

Instructor maintains the name of the target window in the *targetWindow* property. For example, if a handler executes the *draw* command, Instructor draws a new object on the page in the target window. Likewise, if you execute the statement *get text of field "customer",* Instructor searches for the field *customer* only on the page displayed in the target window.

Instructor makes a distinction between the target window and the focus window, which is the viewer containing the page that currently accepts keyboard input. In most cases, the target window is the same as the focus window. For example, if a user clicks the main window, it becomes the focus window (where the user can type text or press access keys), and it also becomes the target window for OpenScript commands. However, there are some situations where the target window is not the focus window. For example, the *mouseEnter* event may be sent when the pointer is over an object that is not in the focus window.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

You cannot explicitly change the value of the *targetWindow* property, but you can control which viewer is the target window by:

- clicking a window to activate it.
- changing the target window temporarily in a script, which allows you to direct some commands and object references to a different window by using the *in <viewer reference>* control structure.
- sending a message using the *in <viewer reference>* parameter of the *send* command to an object in another window.

You can also use explicit object references to control which page, background, and book Instructor searches when getting or setting property values or handling messages.

Activating viewers

You can activate a viewer by clicking it anywhere. When you click a viewer, Instructor makes it both the focus window and the target window. If the viewer is a pop-up viewer (or the main window), its caption bar is highlighted.

You can also activate a pop-up viewer (but not a child viewer) using the *activate* command:

```
activate viewer "Map"
```

If a child window of the pop-up viewer had the focus the last time the pop-up viewer was activated, the *activate* command puts the focus back into that child viewer. For example, if the focus was in a toolbar that is the child of a pop-up viewer, the toolbar again receives the focus when you execute the *activate* command to return to the pop-up viewer.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

To change the focus to a child viewer, set the *focusWindow* property to a viewer reference:

```
focusWindow = viewer "toolPalette"
```

Setting the *focusWindow* differs from using the *activate* command because you can control which child viewer gets the focus. When you use *activate*, Instructor sets the *focusWindow* to the child viewer that had the focus when its parent was previously activated.

Use the *activate* command to switch between pop-up viewers while retaining the focus of their child windows, and use *focusWindow* to change the focus without regard to its previous setting.

Unlike clicking the viewer with the mouse, setting the *focusWindow* property does not change the focus and the target windows at the same time. Instructor changes the focus window immediately but does not change the target window until Instructor reaches the idle state again; until then, the target window is still the viewer it was before the *focusWindow* property was changed.



Tip You can show a viewer but not activate it by executing the *show* command with the *notActive* parameter, which is useful for displaying tool palettes.

Changing the target window temporarily in scripts

Use the *in* control structure to change the target window temporarily while a script is running. The *in* control structure is useful when you want to run commands that affect a viewer other than the target window—for example, when you want to draw objects or flip pages in another viewer.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

All statements contained in the *in* control structure execute in the context of the specified viewer. When the *in* control structure ends, Instructor sets the target window back to the viewer that was the target window before the control structure was executed. For example:

```
to handle buttonClick
   --Changes the target window so that commands and
   --object references refer to the page of a different viewer
   system svCurrentCountry
   in viewer "Map"
      hide statusBar
      text of field "Country" = svCurrentCountry
   end in
   --Target window reverts to its previous value here
end buttonClick
```

Changing the target window using the *send* command

You can specify a new target window for a message by adding the *in viewer* clause to the *send* command. After the *send* command completes execution, including any handlers that it triggers, *targetWindow* is restored to its previous value. For example:

```
send buttonClick to button "Cancel" in viewer "Map"
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Referring to objects in the target window

If you do not include a page reference when referring to an object in a script—that is, if you use an implicit object reference—Instructor searches for the object on the page displayed in the target window. For example, the following three object references are the same; the first one is an implicit object reference, and the second two are equivalent explicit references:

```
--Implicit reference
get caption of button "Start"
--Explicit references
get caption of button "Start" of page "Introduction" of book "myapp"
get caption of button "Start" of currentPage of targetWindow
```

Instructor also interprets the word *this* as referring to the target window. The reference *this page* refers to the page currently displayed in the target window, and *this background* refers to the background that owns the page in the target window. The reference *this book* refers to the book that owns the page displayed in the target window; if the page being displayed is from another book, *this book* refers to that other book, not the one represented in the main window. For example:

```
--Displays name of page displayed in a specific viewer
to handle rightButtonUp
   in viewer "Map" of book "Geography"
      get name of this page
      caption of statusBar = It
   end in
end rightButtonUp
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

To get or set properties for objects that are not in the target window, use an explicit object reference:

```
get text of field "Lesson" of page "Test" of book "C:\Book\tutorial.tbk"
```

Closing the target window

If you close the target window in a handler, the viewer becomes unavailable, and you cannot set any of its nonpersistent properties. However, it is still the target window, and you can use implicit references to objects to get or set their properties until the handler completes execution.

For example, if you are displaying a dialog box in a viewer, the handler for the OK button might close the viewer. However, the handler can contain statements that reference objects on the page after the viewer is closed:

```
--Script for OK button in dialog box viewer
to handle buttonClick
   system dlgResult                          --System variable stores
                                             --user input
   dlgResult = text of field "user name"     --Gets user input
   close targetWindow                        --Closes viewer
   clear text of field "user name"           --Clears user input
end
```



Tip To avoid errors and make debugging easier, do not work with the current page of a closed viewer unless absolutely necessary. Whenever possible, close a viewer with the last line of a handler.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Changing a viewer's parent window

By default, a viewer's parent window is the window that was the target window when the viewer was opened. However, you can make any window the parent of another, so you can nest viewers. For example, you can create a toolbar and assign it to a pop-up viewer. To change a viewer's parent window, set its *parentWindow* property to a reference to the new parent. For example, the following statement makes viewer *Map* the parent of viewer *Legend*:

```
--Viewer "map" must be open
parentWindow of viewer "legend" = viewer "Map"
```

If you close viewer *Map*, viewer *Legend* closes as well.



Note Changing a viewer's parent window does not affect its parent object, which is the object to which the viewer forwards messages. The parent object for all viewers is always the book.

You can set a viewer's *parentWindow* property to *null*, which makes the Windows desktop its parent. The viewer whose parent window is the desktop is then an overlapped pop-up window like the main window. You can overlap the main window on top of it, and it appears in the Windows task list. It can also appear when Instructor is minimized, which is useful for utilities that are available even if Instructor is minimized, such as clocks, calendars, and calculators. For example, the following statement makes the Windows desktop the parent of a viewer named *Clock*:

```
parentWindow of viewer "Clock" = null
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using window handles

In addition to changing a viewer's *parentWindow* property, you can change a viewer's relationship to other viewers and to the Windows desktop by manipulating its handles, which are unique identifiers for that instance of the viewer. Every open viewer has two unique identifiers: a window handle and a client handle. These identifiers are never the same for two different open viewers. A viewer's *windowHandle* identifies its frame window, while the viewer's *clientHandle* identifies its client window. Both properties are nonpersistent; if you close and reopen a viewer, its handles change. You can view these properties but cannot set them.

The viewer's parent window is identified by its *parentHandle* property, which contains the parent window's window handle. You can change the viewer's parent by setting its *parentHandle* property to the window handle of any other window. You can use a viewer's window handle to obtain its complete reference using the *windowRefFromHandle( )* function. For example, if you want to know the viewer reference of the target window, you can use the following statement:

```
to get lastActiveWindow
   get activeWindowHandle
   --Doesn't change if the viewer has revertFocus set to true
   return windowRefFromHandle (It)
end lastActiveWindow
```



Tip Advanced users familiar with Windows programming can get the handle of a window in an application other than Instructor by using the Windows *findWindow( )* DLL function in User.exe.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Creating dialog boxes with viewers

You can use a viewer to display a dialog box. Because viewers display pages, you must draw any controls or fields that you want to appear in the dialog box on a page and then show that page in a viewer. The viewer that you create should appear and behave like a standard Windows dialog box. The best way to ensure this consistency is to use the dialog box viewer template to create the viewer. For details on using viewer templates, see “Creating viewers,” earlier in this chapter.

You can take two different approaches when creating a viewer to display a dialog box. You can create a new viewer for each dialog box you want to display, or you can create a single viewer and display different pages in it as different dialog boxes. Using a new viewer for each dialog box is somewhat easier, but using one viewer for all dialog boxes is more efficient.

After you open a dialog box viewer, you usually initialize its controls. For example, you might want to check certain check boxes or radio buttons, set the text of various fields, highlight a specific item in a list box, or set the focus on a particular button. If you use one viewer per dialog box, you can initialize the controls in the viewer’s script, because you know exactly what page the viewer displays. If you use a single viewer for all dialog boxes, you should initialize controls in the page script.

The following two sections outline both approaches for creating a viewer to display a dialog box.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Creating a new viewer for each dialog box

The example in the following procedure shows you how to create a viewer that functions as a dialog box. It involves creating a viewer, drawing the viewer controls on a page, initializing the dialog box, and retrieving a value that the user typed into the dialog box.

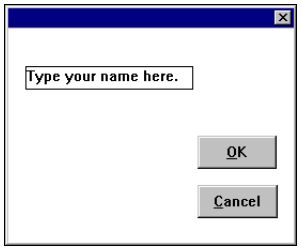


Figure 10
A viewer that displays a dialog box

➤ To create a viewer for displaying a dialog box:

1. From the Insert menu, point to New Viewer, and then choose Dialog Box.
2. Go to the new page that Instructor created for this viewer (the last page in the book), and then name it *dialog page*.
3. From the Object menu, choose Viewers.
4. Select the viewer you just created from the list, and then click Properties.

   Instructor displays the Properties for Viewer dialog box.
5. In the Name box of the Properties for Viewer dialog box, type *dialog viewer*.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

6 Click the Edit Script button on the toolbar in the Properties for Viewer dialog box, and then place the following handler in the script of the viewer:

```
to handle openWindow
   --Sets text of field to value based on a user property;
   --init is a user property
   text of field "text field" = my init
   forward
end openWindow
```

7 On page *dialog page*, draw two buttons named *OK* and *Cancel*, and then draw a text field named *text field*.

8 Place the following handler in the script of the *OK* button:

```
--Hides the viewer without closing it, which indicates to the
--calling program that the user clicked OK
to handle buttonClick
   --Assigns contents of text field as return value of the
   --dialog box
   retValue of targetWindow = text of field "text field"
   hide viewer "dialog viewer"
end
```

9 Place the following handler into the script of the *Cancel* button:

```
to handle buttonClick
   close viewer "dialog viewer"
end
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

10 On the first page (page ID 0) of the book, create a button with the name *Start* and enter the following handler into its script:

```
to handle buttonClick
   --Sets initial value of text field by assigning it to a
   --user property
   init of viewer "dialog viewer" = "Type your name here."
   show viewer "dialog viewer" as modal
   if isOpen of viewer “dialog viewer” then --User clicked OK
      retValue = retValue of viewer "dialog viewer"
      request retValue                                --Displays return
                                                      --value
   end if
end buttonClick
```

11 Switch to Reader level (press F3), and then click the Start button.

The viewer *dialog viewer* will appear, and the text field will contain the text *Type your name here*.

Type your name in the text field, and then click OK. A dialog box will appear containing your name, which is the value that was returned from the text field in the viewer *dialog viewer*.

Creating one viewer to display any dialog box

You can also use a single viewer to display different dialog boxes. You create a single viewer and a number of pages, each of which contains different information to display in the dialog box. To display a dialog box, open the viewer, and then set its current page and the default values of objects in it. To simplify the process, the example provided in the following procedure uses a user-defined function that you can call with different parameters to display different pages in the viewer.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To create one viewer that displays any dialog box:**

1 Open a new book and enter the following handler into its script:

```
--User-defined function with three parameters to show a dialog box
to get showDialogBox pPage, pCaption, pinitString
   local retValue
   dlgViewer = viewer "dialog viewer"
   --Opens viewer to be used as the dialog box, sets page to display
   defaultPage of dlgViewer = pPage
   open dlgViewer
   clear defaultPage of dlgViewer
   caption of dlgViewer = pCaption
   --Sends initialization string to the page
   --in dlgViewer
   send initDialog pinitString to currentPage of dlgViewer
   show dlgViewer as modal
   if isOpen of dlgViewer is true then
      --User clicked OK
      retValue = retValue of dlgViewer
      close dlgViewer
   end if
   return retValue
end showDialogBox
```

2 Create a new page named *dialog page*. Enter the following handler into its script:

```
to handle initDialog pinit
   --Sets text of field to initialization value
   text of field "text field" of self = pinit
end initDialog
```

3 Create a viewer using the dialog box template, name it *dialog viewer,* and set its default page to *dialog page*.

4 On page *dialog page*, draw two buttons named *OK* and *Cancel*, and then draw a text field and name it *text field*.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

5 Place the following handler into the script of the *OK* button:

```
to handle buttonClick
   if targetWindow <> mainWindow
      retValue of targetWindow = \
         text of field "text field"
      hide targetWindow
   end if
end buttonClick
```

6 Place the following handler into the script of the *Cancel* button:

```
to handle buttonClick
   if targetWindow <> mainWindow
      close targetWindow
   end if
end buttonClick
```

7 On the first page (page ID 0) of the book, create a button named *Start* and enter the following handler into its script:

```
to handle buttonClick
   showPage = page "dialog page"
   dialogCaption = "This is my dialog box"
   dialogText = "An error has occurred"
   get showDialogBox(showPage, dialogCaption, dialogText)
   request It
end buttonClick
```

8 Switch to Reader level (press F3), and then click the Start button.

A dialog box will appear with the caption *This is my dialog box*. The text field will contain the text *An error has occurred*.

Type your name in the text field, and then click OK. A dialog box will appear containing your name, which is the value that was returned from the text field in the viewer *dialog viewer*.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

To display other pages in the dialog box, create new pages with OK and Cancel buttons (as on the page *dialog page*), a text field, and other controls as required. Then call the user-defined function *showDialogBox( )*, passing it the name of the new page, a different caption, and different text in the parameters.

Creating a floating tool palette

You can use a viewer to display a floating tool palette that contains graphic objects (such as rectangles and polygons) that you can drag from the palette and drop onto the main window.

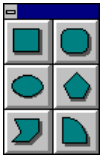


Figure 11
A viewer displaying a tool palette

➤ **To create a floating tool palette:**

1. From the Insert menu, point to New Viewer, and then choose Palette.
2. Navigate to the new page Instructor created for your palette (the last page in the book).
3. Draw controls (for example, a rectangle, a rounded rectangle, a circle, and a five-sided polygon) on the page.
4. In the Properties for Viewer dialog box, on the Behavior tab, be sure that the Always Reader level option is selected.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

5 Enter the following handler into the script for the page displayed in the palette viewer:

```
to handle buttonDown
   drag target silently
end

to handle endDrag dropObject, loc
   local graphic newObject
   --Checks to see if you're dropping onto a page or a background
   if dropObject <> null
      while object of dropObject is not in "page, background"
         dropObject = parent of dropObject
      end
      if (dropObject = currentPage of mainWindow) or \
         (dropObject = parent of currentPage of mainWindow)
         --Decide whether to add the object to the
         --page or to the background
         if onBackground of mainWindow = true
            if object of dropObject = "page" then
               dropObject = parent of dropObject
            end if
         else
            dropObject = currentPage of mainWindow
         end if
         syslockscreen = true
         --Copies object from palette
         newObject = copyObject(target, dropObject)
         --Determines drop position in main window
         in mainWindow
            move newObject to mousePosition of mainWindow
         end in
      end if
   end if
end
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

6 Navigate to one of the pages in the main window.

7 From the Object menu, choose Viewers.

8 Select the palette viewer you created in the list, and then click Show Viewer.

9 Click one of the graphic objects in the viewer, drag it onto the main window, and drop it.

The object will appear in the main window.

Creating a progress bar

You can use a viewer to display a graphical bar, like a thermometer, to track the progress of an event. In the procedure on the following page, you will create a background for a book with ten pages and use the progress bar to display the percentage of activity that is complete as Instructor flips through all of the pages in the book.

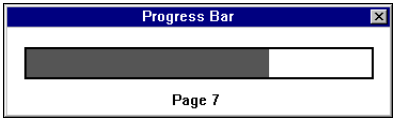


Figure 12
A viewer displaying a progress bar

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To create a progress bar:**

1 Open a new Instructor application, and then create ten new pages.

2 On the first page, create a button named *Prog,* and then draw a graphic object (such as a polygon). Draw a graphic object on each of the remaining nine pages of the book.

Because this application will flip through all of the pages in the book, place a different graphic object on each page so that you can see when the pages are flipping.

3 From the Insert menu, point to New Viewer, and then choose Dialog Box.

4 Navigate to the new page Instructor created for your dialog box. In the Properties for Page dialog box, name the page *viewerPage*, and clear the Skip this page when navigating check box.

5 Draw a rectangle on page *viewerPage* named *fullBar* to represent the full length of the progress bar.

Set the *strokeColor* of *fullBar* to *black* and its *fillColor* to *white*.

6 Create another rectangle on page *viewerPage* named *currentBar* on top of rectangle *fullBar*. Set the *fillColor* of rectangle *currentBar* to *red*. The rectangle *currentBar* will represent the current progress as Instructor flips through the pages in the book.

7 Just below the two rectangles, create a field named *currentPage*. This field will display the current page number as Instructor flips through the pages in the book.

8 Group the two rectangles and the field, and name the group *bar*. Then resize the background to the minimum size needed to accommodate the group.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

9 Enter the following handler into the script of the group *bar*:

```
--Sets the text of the field and the size of the progress bar
to set percentage to pPercentage
   --Checks to see if the percentage passed is a real number
   if isType("real", pPercentage)
      --Checks bounds of the pPercentage parameter
      if pPercentage > 100
         pPercentage = 100
      else
         if pPercentage < 0
            pPercentage = 0
         end if
      end if
      --Sets text of currentPage field to reflect page
      --displayed in the main window
      text of field "currentPage" of self = \
         "Page" && pageNumber of currentPage of mainWindow
      --Stores the width of full progress bar
      fullWidth = item 1 of size of rectangle "fullBar" of self
      --Sets the width of current progress bar
      item 1 of size of rectangle "currentBar" of self = \
         fullWidth * (pPercentage / 100)
   end if
end percentage
```

10 In the Viewers dialog box, select the viewer you created in step 3 (if you started with a blank book, it will be called *Viewer id 1*), and click Properties. Name the viewer *Progress*, and then enter *Page "viewerPage"* as its default page. For details about viewer templates, see "Creating viewers," earlier in this chapter.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

11 Enter the following as the script for the button *Prog* that you created on the first page of the book:

```
to handle buttonClick
   sysCursor = 4    --Changes cursor to hourglass
   open viewer "Progress"
   in viewer "Progress"
      percentage of group "bar" = 0
   end in
   show viewer "Progress"
   flip all pages
   close viewer "Progress"
   --Restores cursor
   sysCursor = default
end buttonClick
```

12 Enter the following handler into the book script:

```
to handle enterPage
   --Ensures that the book is at Reader level
   if sysLevel is "reader" and isOpen of viewer "Progress" = true
      --Sets progress percentage based on the current page number
      percentage of group "bar" of currentPage of viewer \
         "Progress" = ((pageNumber of this page)/(pageCount \
         of this book)) * 100
   end if
forward
end enterPage
```

13 Go to the first page of the book (the page that contains the button *Prog*). Switch to Reader level (press F3), and then click the Prog button. Instructor will flip through all of the pages in the book and show the progress in the Progress viewer.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Creating an application startup screen

You can use a viewer to create an application startup—or splash—screen. Splash screens usually appear for a few seconds when you first start an application. Splash screens often show a corporate logo or an application logo and contain information such as the application's title, version number, and the name of the person to whom the software is registered. The following procedure shows how you can create a splash screen that appears for three seconds every time the user opens a particular book.

➤ **To create an application startup screen:**

1. From the Object menu, choose Viewers.
2. In the Viewers dialog box, click New.
3. In the New Viewer dialog box, choose Use template, and then select Read-only pop-up from the list. Click OK.
4. In the Viewers dialog box, click Properties.
5. In the Properties for Viewer dialog box, name the pop-up viewer *logo*, and then specify the following on the Style tab:
   - Under Caption bar, select None.
   - Under Border style, select Thin frame.
   - Under Visible components, clear the Scrolling check box.

   For details about setting properties of a viewer, see "Setting viewer properties," earlier in this chapter.
6. Navigate to the new page Instructor created for your dialog box and name it *splash*.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

7 Import a graphic onto the page *splash* by choosing Graphic from the Insert menu. Navigate to and select the graphic that you want to appear on the splash screen, and then click OK.

8 Size page *splash* to the dimensions of the graphic.

9 Enter the following handler into the book's script:

```
to handle enterApplication
   show viewer "logo"
   --In a real application, the pause command would be replaced
   --with initialization commands
   pause 3 seconds
   close viewer "logo"
   forward
end enterApplication
```

10 Save and close the book, then open it again, or type *send enterApplication* in the Command window.

The splash screen will appear for three seconds.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Chapter 20

Creating and modifying menu bars

Menu bars, which provide users access to commands in your application, are important sources of functionality and interactivity for your courseware. This chapter describes how to create a menu bar in ToolBook II Instructor using the Menu Bar Editor and OpenScript.

IN THIS CHAPTER

About menu bars **550**

Using the Menu Bar Editor **556**

Working with menu bars **559**

Using OpenScript with menu bars **562**

Handling menu-event messages **576**

Displaying pop-up menus **582**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

About menu bars

A menu bar is an interface element that allows the user to choose commands. Generally, the menu bar runs horizontally along the top of an application's main window. A menu is a list of related commands. For example, the File menu generally contains commands such as Open, Save, and Print.

When you create a menu bar, its information is stored as a menu bar resource that can be assigned to any viewer, including the ToolBook II Instructor main window. When you open a new book, it includes the built-in, Author-level menu bar and a default Reader-level menu bar that is stored as an Instructor resource (*menuBar ID 100*). A resource is an object that can be used multiple times in an application. For details about resources, see Chapter 17, "Using resources."

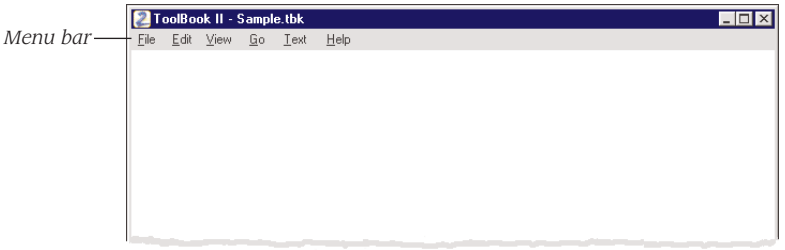


Figure 1
Default Reader-level menu bar

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

A command—or menu item—can have a name and an alias. The name is the text that is displayed on the menu. The alias is a hidden name for the menu item; it is never displayed on the menu, but it represents the menu item in a script. Each built-in Instructor menu item has both a name and an alias. When you add a new menu or menu item, you must give it a name, and you have the option to specify an alias (Instructor does not automatically assign its alias). When a user chooses a menu item, Instructor sends a menu-event message that corresponds to the alias, or to the name if no alias is assigned to the menu item.

Using menu bars on the Internet

When you distribute a native ToolBook II application to be displayed in a Web browser using the Neuron browser plug-in, the menus in the book will function on the Internet just as you authored them in Instructor. If you export your application as a series of Web pages that utilize DHTML, menu bars you authored in Instructor will not appear in your Web page. For details about what to expect when exporting your application as a series of Web pages, see Chapter 24, “Distributing applications on the Internet.”

Types of menus

There are three types of Windows menus that you can create in Instructor: drop-down menus, submenus, and pop-up menus.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

A drop-down menu is the most common type of menu. Drop-down menus appear when a user clicks a menu name; these menus appear in most software user interfaces.

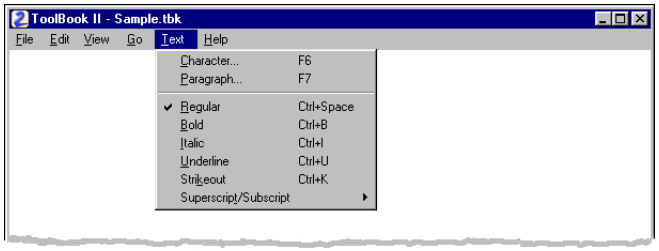


Figure 2
Drop-down menu

An arrow pointing to the right on a drop-down menu command indicates that the command includes a submenu. When you point to the command, the submenu appears at the command's side, as Figure 3 shows. Submenus help simplify and shorten menus and make menu browsing easier. For example, the Superscript/Subscript menu command displays a submenu where you can choose between normal script, superscript, or subscript. You can use several levels of submenus to create a cascading menu.

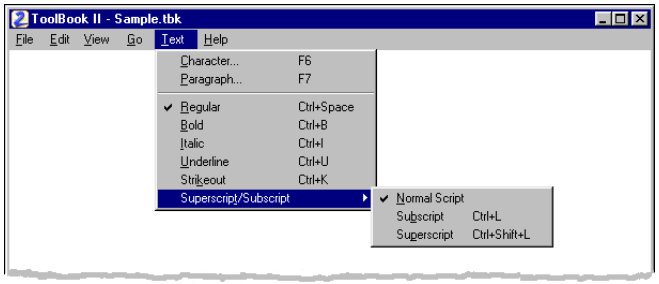


Figure 3
Drop-down menu featuring a submenu

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

A pop-up menu is a context-sensitive menu that is not attached to a menu bar (an example is shown in Figure 4). You can display a pop-up menu anywhere within an Instructor viewer, including the main window. For example, when you right-click an object at Author level, Instructor displays a custom pop-up menu, called the right-click menu, for that object. For details about creating pop-up menus, see "Displaying pop-up menus," later in this chapter.

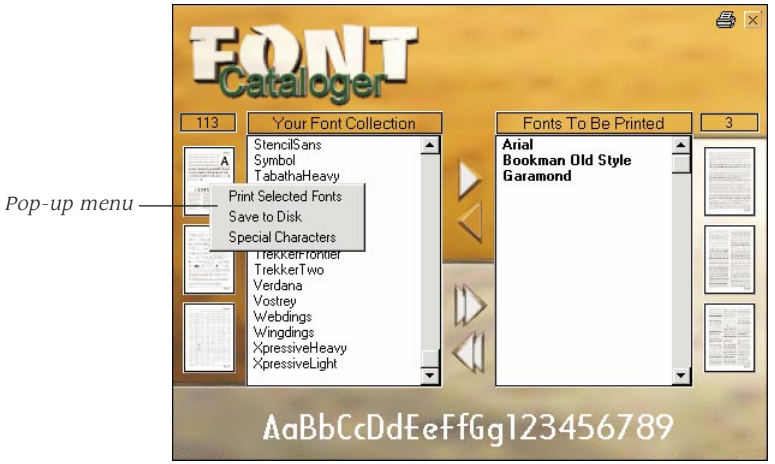


Figure 4
Pop-up menu

Tools for creating and editing menu bars

You can create and edit menu bars in Instructor using the Menu Bar Editor, OpenScript, or some combination of the two. The table on the following page outlines when you can use the Menu Bar Editor and when you should use OpenScript.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

The Menu Bar Editor is a tool included with Instructor that allows you to create levels of menus, menu items, and submenus so that you can customize the user interface. It features a preview window to help you test your menu structure. For details about the Menu Bar Editor, see "Using the Menu Bar Editor," later in this chapter.

Menus rely on OpenScript commands to determine how they will behave. You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book. For more information about using OpenScript to define the appearance and behavior of menus, see "Using OpenScript with menu bars," later in this chapter.

When to use the Menu Bar Editor or OpenScript

To do this...	Use...
Create a new menu bar	Menu Bar Editor
Create a new menu bar for a viewer	Menu Bar Editor
Create a drop-down menu	Menu Bar Editor or OpenScript
Create a submenu	Menu Bar Editor or OpenScript
Display a pop-up menu	OpenScript
Modify an Author-level menu bar*	OpenScript
Modify a Reader-level menu bar	Menu Bar Editor or OpenScript
Modify a menu bar for a viewer	Menu Bar Editor or OpenScript
Modify a menu bar while application is running; for example, check or uncheck menu items, disable or enable menus or menu items, add or remove menus or menu items	OpenScript

* The Author-level menu is not a menu bar resource, so it cannot be modified in the Menu Bar Editor. All other menus are resources.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

How menus work

When a user chooses a menu item, Instructor sends either one or two OpenScript messages:

- First, Instructor sends the *menuItemSelected* notification message with the name and alias of the menu item.
- If there is no handler for the *menuItemSelected* message, or the *menuItemSelected* message is forwarded to the system, Instructor sends the name or alias of the menu item as a menu-event message.
- If a handler for the *menuItemSelected* message contains a *forward to system* statement, Instructor sends both the *menuItemSelected* message and the menu-event message for the specific menu item.

A menu-event message corresponds to a menu item's name or to its alias, if one exists. For example, when a user chooses Open from the File menu, Instructor sends the *open* menu-event message, which is the built-in alias for the Open menu item. If you created a new menu item named Options, Instructor would send *options* as its menu-event message. If you assigned an alias to the Options menu item, such as *listOptions*, Instructor would send that alias instead of *options*.

Instructor provides automatic behavior for built-in menu-event messages, but you must write handlers for any menu-event messages that you define. You can also use or alter the behavior of a built-in menu item by writing a handler for a built-in menu item alias.

For details on writing handlers for *menuItemSelected* or a specific menu-event message, see "Handling menu-event messages," later in this chapter.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using aliases for menus and menu item names

When you use aliases, you can use any name for a menu or menu item and create its behavior by writing a handler for its alias or by using a built-in Instructor alias. Instructor's built-in aliases for menu items are referred to as menu-event messages in the OpenScript reference in online Help.

You can change menu and menu item names while their aliases and behavior remain intact. This capability is especially useful for situations in which you are preparing an application that will be translated into several languages. Aliases can be used in several different ways. You can:

- assign your own alias to a new menu or menu item and write a handler to create its behavior.
- assign a built-in Instructor alias to a new menu item to use a built-in menu item's default behavior.
- write a handler for a built-in Instructor alias to modify a menu item's default behavior.

Using the Menu Bar Editor

One of the ways you can create and edit menu bars is to use the Menu Bar Editor. When you create a new menu bar and then save the book, the menu bar is saved as a resource of that book. Using the Menu Bar Editor, you can create any number of menu bar resources for a given book and then assign one to a viewer or the Reader-level main window.

The Menu Bar Editor works like an outliner. You can create multiple levels of menus, menu items, and submenus by promoting and demoting text. You can test your menu in the Menu Bar Preview window.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

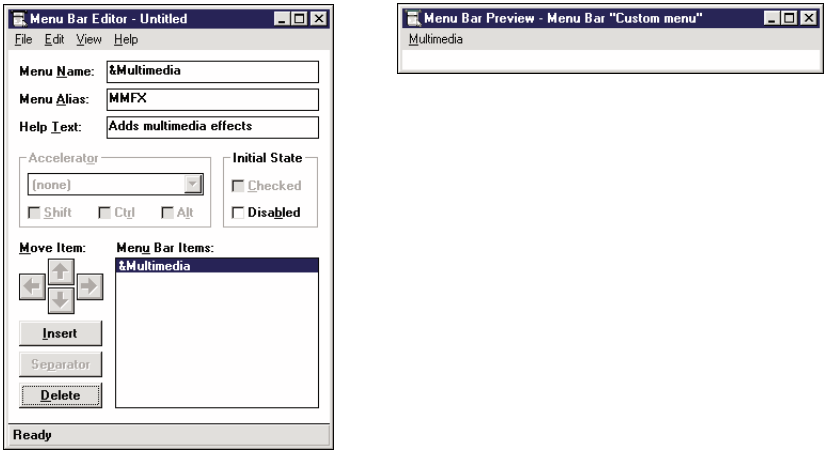


Figure 5
The Menu Bar Editor and the Menu Bar Preview window

➤ **To create and edit a menu bar using the Menu Bar Editor:**

1 From the Object menu, choose Resources.

Instructor displays the Resource Manager dialog box.

2 Select Menubar in the Available resources list, and then click New.

Instructor displays the Menu Bar Editor.

3 To add a menu or menu item, type the item's name in the Menu name box, and then press ENTER.

Tip To use one of the letters in the menu or menu item as an access key, type an ampersand (&) before the letter. For example, if you type *Te&xt*, *x* is the access key; the user can press ALT in combination with the access key "x" to gain access to the menu or menu item. To make an ampersand appear in a menu or menu item, use two ampersands (&&).

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

4 To create multiple levels of menus, menu items, and submenus, you promote, demote, move, delete, or add items:

- To promote a menu item, select it, and then click the left arrow button that appears under Move item.
- To demote a menu item, select it, and then click the right arrow button that appears under Move item.
- To move a menu item up or down in the list, select it, and then click the up or down arrow buttons that appears under Move item.
- To add a separator bar between menu items, click Separator, or insert a menu item with no text for its name.
- To delete a menu item, select it, and then click Delete.
- To insert a menu item, select the item that you want to appear just below the new item, and then click Insert. Instructor inserts a blank menu or menu item before the selected menu or menu item.

5 To set options for a menu or menu item, select it, and then select or enter one or more of the following:

- **Menu alias** Specifies the message to be sent in place of the name.
- **Help text** Sets the text to be displayed in the status bar.
- **Accelerator** Specifies the shortcut key combination (sometimes called the accelerator) for the menu item.
- **Checked** Displays a check mark next to the menu item by default.
- **Disabled** Disables (dims) the menu or menu item by default.

6 If the Menu Bar Preview window is not open, from the View menu, choose Menu Bar Preview. Test your menu bar by choosing menus and menu items.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

7 To add the menu to the book, from the File menu, choose Update. You are prompted to name the resource.

8 Close the Menu Bar Editor.

The menu bar resource appears in the Resource Manager with its ID number.



Note To save new or modified menu bar resources with a book, be sure to save the book you used to create or edit those resources.

The procedure above describes how to create and edit a menu bar. In order to view a menu bar in your application, you must first assign it as a resource to a viewer. The following section details how to include a custom menu bar in your book.

Working with menu bars

When you create a menu bar, Instructor automatically creates a resource that you can then assign to a viewer (or multiple viewers). Unlike other resources, menu bars are specific to Instructor and cannot be used in other applications. As with any Instructor resource, you can export, import, copy, replace, or remove menu bar resources. For details about resources, see Chapter 17, “Using resources.”

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using menu bar resources in a book

In a new book, the main window's default Reader-level menu bar is referenced by the ID number *menuBar 100*. You can customize the Reader-level menu bar either by editing *menuBar ID 100* or by assigning a different menu bar to the main window.

You assign a menu bar to a viewer by specifying the menu bar resource in the viewer's Properties dialog box or by setting the viewer's OpenScript *menuBar* property to a menu bar resource reference. When you assign a menu bar resource to the main window, it appears as the book's Reader-level menu bar. When you assign a menu bar resource to any other viewer, it appears at both Reader and Author levels.



Note The Instructor Author-level menu bar is not a resource; you cannot replace it with a menu bar resource, but you can modify its menus and menu items using OpenScript. For details, see "Using OpenScript with menu bars," later in this chapter.

Assigning a menu bar resource to a viewer

You can assign a menu bar resource to any viewer. Each viewer has a *menuBar* property that specifies the resource it uses as its menu. You can also assign a menu bar resource to the main window (*viewer ID 0*) for use at Reader level.



Note If you want to assign a menu to a new viewer, create the viewer and then open the Viewers dialog box. For information about creating a viewer, see Chapter 19, "Creating windows with viewers."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ To assign a menu bar resource to a viewer:

Using the interface

1. From the Object menu, choose Viewers.

   The Viewers dialog box appears.

2. In the Viewers of book list, select the viewer to which you want to assign the menu bar resource. If you want to apply the menu to the Reader-level main window, select Main window.

3. Click Properties to display the selected viewer's Properties dialog box.

4. On the Style tab, under Menu bar, click Choose to display the Choose Menu dialog box.

5. In the graphical list, select the menu bar resource you want to use, and then click OK to close the Choose Menu Bar dialog box.

6. Close the Properties for Viewer dialog box.

7. Close the Viewers dialog box. Instructor automatically sets the *menuBar* property to the menu bar resource you selected.

Using OpenScript

- To assign a menu bar to a viewer or the main window at Reader level (*viewer ID 0*), set the *menuBar* property to a menu bar resource:

```
menuBar of viewer "tools" = menuBar "custom"

--Assigns menu bar to main window at Reader level
menuBar of viewer MainWindow = menuBar "userMenu"
```



Note Menu bars will appear only on viewers that have the caption bar set to Normal or None, and the Border style set to Thin frame or Thick frame. A menu bar will not appear on a viewer with any other frame properties.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Removing a menu bar resource from a viewer

You can remove a menu bar resource from a viewer using the viewer's Properties dialog box or OpenScript.

➤ **To remove a menu bar resource from a viewer:**

Using the interface

1 Open the viewer's Properties dialog box.

2 On the Style tab, under Menu bar, click Clear.

Using OpenScript

◆ To remove a menu bar resource from a viewer, including the main window's Reader-level menu bar, set the viewer's *menuBar* property to *null*:

```
menuBar of viewer "facts" = null

--Restores the default menu bar resource for the main window
--at Reader level
menuBar of viewer ID 0 = null
```

Using OpenScript with menu bars

In addition to using the Menu Bar Editor to create and edit menu bar resources, you can also use OpenScript to temporarily modify Instructor's built-in menu bars and viewer menu bars. You can add new menus and menu items or customize existing ones.

You can write scripts to add, change, or replace menu items or the entire menu. For example, to suit varying circumstances in your application, you can check or uncheck menu items, disable or enable menus or menu items, and add or remove menus or menu items.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index



Note When you use OpenScript to modify the menu bar of any viewer, including the main window, that viewer must be the target window. For more information about working with viewers, see Chapter 19, "Creating windows with viewers."

Any modifications made to a menu bar with OpenScript commands, such as *add menu*, *check menuItem*, *remove menu*, and so on, are temporary and cannot be saved with the book; the menu bar is restored to its default settings when the book closes. To make permanent changes to a menu bar, use the Menu Bar Editor. Edits made to a menu bar resource with the Menu Bar Editor can be saved as part of that resource.

Use the OpenScript commands in the following table to modify the built-in menu bars at Author or Reader level in the main window, or any menu bar resource displayed in a viewer that is the target window.

Modifying a menu bar using OpenScript

To do this...	Use this command or function...
Add a menu to an existing menu bar	*add menu*
Add a submenu to an existing menu	*add menu*
Assign an alias to a menu	*add menu*
Add Help text for a menu	*add menu*
Add a menu item to a menu	*add menuItem*
Add a separator bar between menu items	*add menuItem*
Assign an alias to a menu item	*add menuItem*
Add Help text for a menu item	*add menuItem*
Delete a menu from an existing menu bar	*remove menu*
Delete a menu item from a menu	*remove menuItem*

(continued)

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Modifying a menu bar using OpenScript, *continued*

To do this...	Use this command or function...
Restore a menu bar resource or Author-level menu bar to its default setting	*restore menuBar*
Hide a menu bar from view	*hide menuBar*
Show a menu bar if it is hidden	*show menuBar*
Remove a separator between menu items	*remove separator*
Change the name of a menu or menu item	*setMenuName( )* *setMenuItemName( )*
Change the Help text for a menu or menu item	*setMenuHelpText( )* *setMenuItemHelpText( )*
Enable a menu or menu item that was disabled	*enable menu* *enable menuItem*
Disable (dim) a menu or menu item	*disable menu* *disable menuItem*
Add a check mark to the left of a menu item	*check menuItem*
Remove a check mark from a specified menu item	*uncheck menuItem*
Check whether a menu is enabled or disabled	*menuEnabled( )*
Check whether a menu item is enabled or disabled	*menuItemEnabled( )*
Check whether a menu item is checked or unchecked	*menuItemChecked( )*



Note You cannot modify the pop-up menu that appears when you right-click the title bar in Instructor.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

If your application changes the menu bar, you can restore its original settings by using a *restore menuBar* statement in a *leaveApplication* handler. Otherwise, if the book's *keepMenuBar* property is set to *true*, Instructor will display any modifications made to the menu bar in the main window when you navigate to another book.

Adding menus and menu items

You can add up to 60 menus with 255 commands of 60 characters each to your Instructor application. To add a new menu, use the *add menu* command and specify the menu name. You can also specify the menu location, alias, Help text, and working level. For example:

```
add menu "Re&ferences" at reader with \
   helpText "Show the References screen"
```

Adding a menu item is similar to adding a menu. To add a menu item, use *add menuItem* and specify the name of the item you want to add. You can also specify the alias, Help text, and working level. For example:

```
add menuItem "Glossary" to menu "Help" at author \
   with helpText "Help on multimedia functions"
```

For menus that are used throughout a book, place the *add menu* and *add menuItem* statements in the *enterApplication* handler of the book's script. Place these statements lower in the hierarchy if you want different menus for various pages and backgrounds.

The following handler in a book's script adds a menu with three menu items at Reader level:

```
to handle enterApplication
   add menu "Lessons" at reader
   add menuItem "Introduction" to menu "Lessons" at reader
   add menuItem "Basics" to menu "Lessons" at reader
   add menuItem "Advanced Topics" to menu "Lessons" at reader
   forward
end
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

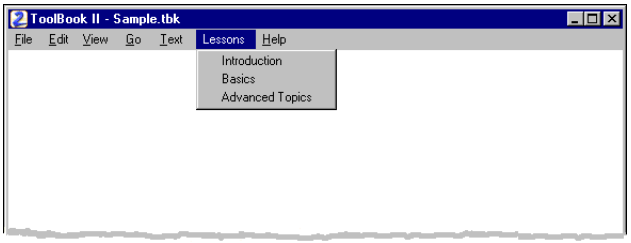


Figure 6
Custom menu

Adding submenus to a menu

You create a submenu by adding a menu to another menu. You can use several levels of submenus to create a cascading menu:

```
--Adds a submenu to ToolBook II Help menu
add menu "Using this application" in menu "Help"
add menuItem "Tips" to menu "Using this application" in menu "Help"
add menuItem "Sample scripts" to menu "Using this application" \
   in menu "Help"
```

You can create a cascading menu by adding a submenu to a submenu:

```
--Adds a submenu to a submenu
add menu "Options" in menu "Using this application" in menu "Help"
```

Adding an access key to a menu or menu item

Access keys, which are the underlined characters in menu and menu item names, provide a way to choose a menu or menu item without using a mouse. For example, to display the Open dialog box, you press ALT+F to drop down the File menu, and then press ALT+O to choose the Open menu item.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

To specify an access key when you add a menu or menu item, add an ampersand (&) before the character you want to serve as the access key:

```
add menuItem "Advanced &Topics" to menu "Lessons"
```

The *T* is underscored on the menu, but the ampersand is not part of the *advancedTopics* message.

Adding a new alias for a user-defined menu or menu item

You can assign an alias to any menu item or custom menu you create (that is, a user-defined menu) by including it in the syntax for the *add menu* or *add menuItem* command. For example, in a tutorial application, you might have a menu called Lessons with two menu items called Next and Previous. To avoid confusing the *next* menu item name with Instructor's built-in *next* alias, you could assign the alias *nextLesson* to the Next menu item and then write a handler for *nextLesson* to define its behavior:

```
--Creates a user-defined menu item Next with an alias nextLesson
add menuItem "Next" alias "nextLesson" to menu "Lessons"
```

If you assign an alias that is not built in to Instructor, you must write a handler to define its behavior. For details, see "Handling menu-event messages," later in this chapter.

You may also want to assign an alias to a menu because a menu's alias is included as a parameter for the *enterMenu* notification message. For details about using *enterMenu*, see "Displaying menu items conditionally," later in this chapter.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Adding a built-in alias for a user-defined menu item

You can use Instructor's default menu item behavior by assigning a built-in alias to a new menu item.

Instructor provides automatic behavior for any alias that corresponds to a built-in menu item. This behavior includes not only Instructor's default response to the message—such as saving changes or going to the next page—but also how the menu or menu item itself behaves within the application. For example, the Cut command is disabled (appears dimmed) on the menu when no object is selected. Other examples of automatic behavior include placing a check mark beside a selected menu item or changing the message that is sent based on the current state of the book.

You might create a menu item called Show Jumps that sends the standard Instructor alias *showHotwords*:

```
add menuItem "Show Jumps" alias "showHotwords" to menu "Options"
```

When a user chooses Show Jumps, the *showHotwords* alias is sent, and the expected Instructor behavior results: A check mark is placed beside the command name, and all hotwords are highlighted.

Instructor's built-in aliases often resemble the name of the menu item. You can usually specify a standard Instructor alias by typing the menu item name you want and omitting spaces between words. For example, the alias for the Save As menu item is *saveAs*.



For details on each built-in Instructor alias, refer to the menu-event message entries in the OpenScript reference in online Help.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Aliases are useful when you are translating your application into another language. You can use aliases instead of menu item names in all scripts and handlers to ensure that the application will continue to work after translation. For example, the following statements use an alias to add a menu item in a French version and in a German version:

```
add menuItem "&Fenêtre1" alias "Window1" to menu "Fichier" at reader
add menuItem "&Fenster1" alias "Window1" to menu "Datei" at reader
```

When you remove a user-defined menu or menu item with either the *restore menuBar*, *remove menu,* or *remove menuItem* command, the alias for that menu or menu item is also removed.

Adding Help text for a menu or menu item

When you use the *add menu* or *add menuItem* command, you have the option to use the *with helptext* <*text*> parameter to assign a string of Help text that appears in the Instructor status bar whenever the user chooses a menu or menu item. For example:

```
add menu "Using this application" in menu "Help" \
   with helpText "Information on how to become a power user"
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

You can also change the Help text assigned to a menu or menu item. You can write a handler using the *setMenuHelpText( )* or *setMenuItemHelpText( )* functions to set the Help text that appears in the status bar for a menu or menu item based on a set of conditions:

```
--Changes the help text for a menu item depending on the state
--of the system variable s_isLastLesson
to handle menuItemSelected mName, mAlias
   system logical s_isLastLesson
   if mAlias is nextLesson
      if s_isLastLesson
         get setMenuItemHelpText(next, "Displays next module")
      else
         get setMenuItemHelpText(next, "Displays next lesson")
      end if
   end if
   forward
end menuItemSelected
```

For details about using the *setMenuHelpText( )* and *setMenuItemHelpText( )* functions, refer to their entries in the OpenScript reference in online Help.

Changing the names of menus and menu items

Use the *setMenuName( )* or *setMenuItemName( )* functions to change the menu or menu item name on a built-in menu bar or a menu bar resource to suit the currently available options in your application. For example, Instructor changes the command name for deleting an OLE object on the Author-level Edit menu depending on the type of OLE object selected.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

You can also use these functions to change the default Author- and Reader-level menu and menu item names. If you change the name of a built-in menu or menu item, the position of the access key remains the same, regardless of which character assumes that position in the name. For example, you can access the Page menu using ALT+P. If you change the name of the Page menu to Where To?, you can access the menu using ALT+W, even though the underscore is not visible.



For details about using the *setMenuName( )* and *setMenuItemName( )* functions, refer to their entries in the OpenScript reference in online Help.

Putting check marks beside menu items

Use the *check menuItem* and *uncheck menuItem* commands to add or remove check marks on menu items. You must also write handlers to define the related behavior:

```
--Place in a book script to sort pages by zip code.
--Shows how the field is sorted by removing the check mark from "name"
--and creating a check mark next to "zipcode"
--The user-defined menu items are created in another handler
to handle sortByZip
   sort by ascending number text of recordField "zip"
   uncheck menuItem "name"
   check menuItem "zipcode"
end
```

Use the *menuItemChecked( )* function to find out whether a menu item is checked:

```
to handle enterPage
   --Show a field if the menu item is checked
   if menuItemChecked("Review") is true
      show field "Summary"
   end if
   forward
end enterPage
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

If you use the *menuItemChecked( )* function to query the state of a menu item that does not exist, an Execution Suspended error results.

Adding and removing separator bars

Use the *add menuItem* command with a null string to add a thin, horizontal line to separate groups of related menu items:

```
--Adds a separator bar after the first menu item
add menuItem null to menu "View" position 2

--To remove a separator bar, use the remove separator command
remove separator 1 in menu "View" at position 2
```

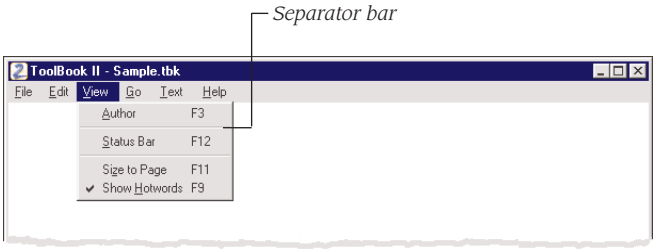


Figure 7
Menu featuring separator bars

Enabling and disabling menus and menu items

You can control the state of menus and menu items with the *enable* and *disable* commands. When you create a menu or menu item in the Menu Bar Editor, you can set its default state. An enabled menu or menu item appears black; a disabled menu or menu item is dimmed, or gray. For example, on the Instructor Author-level Edit menu, the Delete command appears disabled when no object is selected.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

To disable a menu, use the *disable menu* command:

```
disable menu "Expert"
```

The *enable menu* command restores a disabled menu:

```
enable menu "Expert"
```

To disable a menu item, use the *disable menuItem* command. For example, a menu item such as Copy is enabled only when an object is selected; otherwise, it is disabled. The *enable menuItem* command restores a disabled menu item. For example:

```
--Enables the Verify menu item only at Author level
disable menuItem "Verify" of menu "Options" at reader
enable menuItem "Verify" of menu "Options" at author
```

Use the *menuEnabled( )* and *menuItemEnabled( )* functions to find out whether a menu or menu item is enabled or disabled. If the function returns *true*, the menu or menu item is enabled. If it returns *false*, the menu or menu item is disabled.

If you use the *menuEnabled( )* or *menuItemEnabled( )* function to query the state of a menu or menu item that does not exist, Instructor displays the Execution Suspended message.



For details about using the *menuEnabled( )* and *menuItemEnabled( )* functions, refer to their entries in the OpenScript reference in online Help.

Hiding and showing menu bars

The *hide menuBar* statement hides the menu bar. To display the menu bar, use the *show menuBar* statement. You might choose to hide the menu bar during operations such as animation sequences.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

You cannot choose menu items when the menu bar is hidden, but you can still press access keys or use the Command window to send the menu-event messages that correspond to the menu items.

Removing menus and menu items

You can remove entire menus and menu items. To remove menus without removing the menu bar, use the *remove menu* command:

```
remove menu "Text" at reader
```

To remove a menu item, use *remove menuItem*. When you remove a menu or menu item from a submenu, include the *in <menu reference>* parameter for every menu included in the cascading menu. For example:

```
remove menuItem "Sample" in menu "Tutorial" in menu "Help"
```

Removing menus and menu items also disables access keys but has no effect on the scripts that control the behavior of menus and menu items. You can still send a menu-event message to run its corresponding handler.

Restoring a modified menu bar to its original state

The *restore menuBar* command restores the menu bar in the target window to the default settings of the resource assigned to that viewer, or to the built-in defaults for the Instructor Author-level menu bar.

When Instructor executes the *restore menuBar* command, any menus and menu items added with OpenScript commands are removed from the menu bar resource in the target window.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Keeping the same Reader-level menu bar between book instances

Use the *keepMenuBar* book property to preserve the appearance of the menu bar in the main window when navigating between book instances. The *keepMenuBar* property helps maintain compatibility between menu bars created in different versions of Instructor.

When you navigate to a different book in the main window, ToolBook displays the menu bar assigned to the main window in the new book. In versions of ToolBook 1.53 and earlier, ToolBook did not change the main window's menu bar upon navigation to a new book; the developer had to use commands to change the menu bar in the new book's script. You can use either behavior by setting the *keepMenuBar* property to *true* or *false*.

To continue displaying the same menu bar between different book instances, set each book's *keepMenuBar* property to *true*.

To ensure that Instructor loads the main window's menu bar regardless of the setting for a previous book's *keepMenuBar* property, add a *restore menuBar at reader* statement to an *enterApplication* handler in a book's script.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Handling menu-event messages

When a user chooses a menu item, Instructor sends the *menuItemSelected* notification message. If the *menuItemSelected* message reaches the system level, Instructor sends a menu-event message corresponding to that menu item's name or to its alias, if one exists.

Instructor automatically creates and sends a menu-event message that corresponds to the menu item's name or alias (if specified), whether it is built in or user defined. Instructor provides automatic behavior for built-in menu-event messages. Whenever you create a new menu item for any menu bar, you must write a script that handles the new name or alias that is sent as a message when the menu item is chosen.

You can write a handler for the specific menu-event message, or you can write a handler for the *menuItemSelected* notification message and include condition statements for individual menu items. In most cases, you can adequately define behavior for new menu items by writing handlers for each menu-event message. Use the *menuItemSelected* message to control menu behavior based on the state of the application.

If you want to define behavior for a drop-down menu just before it is displayed, you can write a handler for the *enterMenu* message. For details, see "Displaying menu items conditionally," later in this section.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Order in which Instructor sends menu-related messages

Message	Target	Result
enterMenu	Viewer that owns menu bar, then the book	None
enterMenu	Viewer that owns menu bar, or the book	Handler for *enterMenu* runs
menuItem Selected	Viewer that owns menu bar, then the book	Instructor sends menu-event message for that menu item
menuItem Selected	Viewer that owns menu bar, or the book	Handler for *menuItem Selected* runs, and no menu-event message is sent
menuItem Selected	Viewer that owns menu bar, or the book	Handler for *menuItem*, *Selected* runs, message is forwarded to system, menu-event message is sent



Note If a handler intercepts a message and does not forward it, the message does not continue up the object hierarchy.

Where to place handlers for menu-event messages

Use this message...	In the script of this object...
Individual menu-event message	Page, background, or book
menuItemSelected	Viewer or book
enterMenu	Viewer or book

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Using the *menuItemSelected* notification message

You can use the *menuItemSelected* notification message to define behavior for menu items that change dynamically, such as a menu item list of the most recently used files.

When a user chooses a menu item, Instructor sends the *menuItemSelected* message to the viewer that owns the menu bar. If the script of the viewer does not handle the message, it is forwarded to the book.

If a handler for the *menuItemSelected* message exists in the script of a viewer or the book, Instructor does not send a menu-event message for the menu item when it is chosen. Instructor sends a menu item's name or alias only when the *menuItemSelected* message reaches the system level.



Note The *menuItemSelected* message will reach the system level if you don't write a handler for it. If you do write a handler for this message, you must include a *forward to system* statement to allow the message to reach the system level. Use *forward to system* instead of *forward* to prevent a system book from interfering with menu-event messages.

To be certain that a handler for the *menuItemSelected* message does not interfere with handlers for other menu-event messages, you should forward the *menuItemSelected* message any time you do not provide specific support for a menu item in the *menuItemSelected* handler.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

The *menuItemSelected* message has two parameters that specify the selected menu item's name and its alias. When you write a handler for the *menuItemSelected* message, you can use a *conditions* control structure with statements that use the <*name*> or <*alias*> parameters to determine which menu item is selected. For example:

```
--Handles menuItemSelected to find out which file is chosen from a
--set of menu items that represent a list of the most recently used
--Instructor files; place in the script of the Main window
to handle menuItemSelected mName, mAlias
   if mAlias contains "file"

      --Sends user-defined message to handle menu item;
      --the menu item aliases are file1, file2, and so on
      send loadFile mAlias
   else

      --A menu item that is not related to the file list was chosen
      forward
   end if
end menuItemSelected
```



For details, about the *menuItemSelected* message, refer to its entry in the OpenScript reference in online Help.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Handling a single menu-event message for a user-defined menu item

If you need to create behavior for only a small number of menu items, you may want to write a handler for each menu-event message. For example, the following handler is executed when the user chooses Advanced Topics from a custom menu:

```
--Place this handler in the script of the page
to handle advancedTopics
   save changes to this book   --Avoids Save Changes dialog box
   go to page "Advanced" of book "c:\lessons\learn.tbk"
end
```

Handling a single menu-event message for a built-in menu item

If you want to modify the automatic behavior of a built-in Instructor menu item, you can write a handler for its menu-event message. For example, you might create a button that a user clicks to navigate to the next page when the *next* menu-event message is sent. To stop navigation on the last page of the book, you could write a handler for the *next* message that displays a message rather than going to the next page. For example:

```
--Place this handler in the script of the last page in the book
to handle next
   request "You have reached the end of this lesson."
end
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

You can also complement a menu item's automatic behavior by handling its menu-event message in the page's script and then forwarding the message so that it continues to use its default behavior. For example:

```
--Handles the save message to display the Request dialog box
--If the user chooses "Yes", the save message is forwarded and uses
--its automatic behavior
to handle save
   request "Are you sure you want to save your changes?" \
      with "Yes" or "No"
   if It is "Yes"
      forward
   end if
end save
```

Displaying menu items conditionally

You can control the appearance of menu items at the time a user selects a menu by writing a handler for the *enterMenu* notification message. You can disable some menu items, enable others, or add check marks before the menu and its items are displayed.

When a user selects a menu, Instructor sends an *enterMenu* message to the viewer that owns the menu bar, just before the drop-down menu is displayed. If this message is not handled by the viewer, it is automatically forwarded to the book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

The *enterMenu* message has two parameters that specify the selected menu's name and its alias. When you write a handler for the *enterMenu* message, you should use a *conditions* control structure with statements that use the <*name*> or <*alias*> parameters to determine which menu is selected:

```
--Checks whether a group is selected before enabling the Edit Group
--menu item on the drop-down menu
to handle enterMenu pMenu, pAlias
   if pAlias = "EditGroup"
      if selection <> null and object of selection = "group"
         enable menuItem "EditGroup"
      else
         disable menuItem "EditGroup"
      end if
   end if
end enterMenu
```



For details about the *enterMenu* message, refer to its entry in the OpenScript reference in online Help.

Displaying pop-up menus

You can display a pop-up menu using the *popupMenu( )* function. You must supply three parameters for the function: a reference to the menu bar resource that contains the menu, the location for Instructor to display the pop-up menu, and the name of the menu to be displayed (if this parameter is *null*, Instructor displays the first menu of the menu bar resource by default). You can display a pop-up menu anywhere within an Instructor viewer, including the main window.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

When you call the *popupMenu( )* function, Instructor displays the specified menu as a pop-up menu at the specified location. When the user chooses a menu item from the menu, the function returns a list containing the text of the menu item and the alias of the menu item. The following example uses a menu bar resource *menuBar "rightClick"*:

```
to handle rightButtonDown pLoc
   get popupMenu(pLoc, menu "rightclick", object of target)
   --The menu items are Instructor messages
   if It is not null
      conditions
         when It = "reshape"
            send reshape target
         when It = "Flip Horizontal"
            send flipHorizontal target
      end conditions
   end if
end rightButtonDown
```



For details about the *popupMenu( )* function, refer to its entry in the OpenScript reference in online Help.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Chapter 21

Using ActiveX, Automation, and other Windows technologies

ActiveX, OLE, Automation, and ActiveX Data Objects (ADO) are technologies you can use to incorporate files, data, or functionality from other applications directly into your ToolBook II Instructor application. This chapter introduces you to some basic concepts and gives instructions for using these technologies.

IN THIS CHAPTER

About ActiveX and Automation **586**

Using ActiveX and Automation in an application **587**

Writing OpenScript for ActiveX controls and Automation objects **591**

Creating action sequences for ActiveX controls **597**

Troubleshooting ActiveX and Automation **600**

About ADO **603**

About OLE **603**

Working with OLE **604**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

About ActiveX and Automation

ActiveX controls are software components that you can use to add custom features and functionality to your application or Web page. An ActiveX control can be simple (such as a calendar) or complex (such as the Microsoft Web Browser control). You can create your own ActiveX control, using another programming environment such as Visual Basic, or purchase one from a software vendor.

When you use an ActiveX control in ToolBook II Instructor, keep in mind that you are adding software that may or may not be designed to work in Instructor. Because many ActiveX controls are built to run over the Internet, you may encounter unusual behavior when you run them in the Instructor environment. Additionally, a control may have dialog boxes that appear to be a part of Instructor but are actually created by the control itself. While some ActiveX controls include documentation that describes the features of the control, many do not.



Note You should test any ActiveX controls you add to your application. For tips on making a control work in Instructor, see "Troubleshooting ActiveX controls," later in this chapter.

Automation (formerly referred to as OLE Automation) provides a mechanism for opening, controlling, and communicating with Automation server applications such as Microsoft Word, Excel, and Outlook. Because it uses Microsoft's Component Object Model (COM) technology, the same underlying technology used by ActiveX controls, you can control and interact with Automation objects in very much the same way that you control and interact with ActiveX controls. Unlike ActiveX controls, Automation objects are not programmable using the Actions Editor. In addition, Instructor's Web export utilities do not support Automation.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

In order for your ToolBook II application to create and interact with an Automation object, the Automation server application must be installed on the same computer. If you distribute your application, your users must have the same Automation server application installed on their computers.

For tips on making Automation objects work in Instructor, see "Troubleshooting Automation," later in this chapter. To see an example of how Automation can be used with Instructor, explore the ToolBook II Instructor Profiler (Samples\Profiler.exe on your Instructor program disc). Profiler is a sample application that demonstrates one use of Automation: using Automation to access and manipulate programmable features in Word. For more information about Profiler and Automation, refer to the Profiler online book (Samples\OLE\Profiler.pdf on your Instructor program disc).

Using ActiveX and Automation in an application

When you add an ActiveX control to your application, it appears as a physical object on a page or background of your Instructor book. Once you draw the control on a page, you can set its properties using the properties dialog box, action sequences, or scripts. You can also respond to a control's events or execute its methods using action sequences or OpenScript. With Automation, no physical object is created; instead, an OpenScript function creates a reference to an Automation object that opens a session with an application's Automation server. You can then use OpenScript to work with the Automation object's events, properties, and methods.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Adding and removing an ActiveX control

In order to use an ActiveX control, you must add it as an extension to your book. Adding a control as an extension tells Instructor that the control will be used in the application and that it should be included when you package and distribute your book. After adding the control as an extension, a new tool for the control will appear on the main tool palette. You can use this tool to draw the control on your page.

Note ActiveX controls export to DHTML. However, they will only function if your application is run in Internet Explorer 4.0 (or later) running Windows.

➤ **To add an ActiveX control to your book:**

1 From the File menu, choose Extensions.



2 In the Extensions dialog box, select an extension from the Available list, and then click the Add button to add the selection to the Added extensions list.

   Only controls that have been installed and registered in the system registry will appear as available extensions. To add a control that has not yet been registered, click Browse to open the Add Extension dialog box, where you can navigate to and select a control to add. Most controls will register themselves at this point. If you have trouble registering a control, refer to the control's documentation.

3 When you have completed your selections, click OK to close the Extensions dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index



When you add an ActiveX control as an extension, the control's tool appears on the tool palette.

Figure 1
The ActiveX control's tool appears on the tool palette

➤ To draw an ActiveX control on your page:

1. After adding the control to your book, from the tool palette, choose the control's tool.
2. Click on your page and hold down the mouse key as you drag to draw the control.

After you have added an ActiveX control to your book, it will remain an extension of your book until you remove it. Deleting a control you have drawn on a page does not remove the extension from your book. You cannot remove an extension if it is used in the book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To remove an ActiveX control from your book:**

1 Delete all instances of the control from your book.

2 From the File menu, choose Extensions.



3 In the Extensions dialog box, select the control you want to remove, and then click the Remove button to remove the selected extension from the Added extensions list.

4 Click OK.

Creating an Automation object

Unlike ActiveX controls, Automation objects are not objects that exist on a page or background; instead, they are created using the createAutoObject() OpenScript function. To create an object reference to an Automation server application, you must pass the Automation server name as a parameter to the createAutoObject() function. Every Automation server has a unique name. Check the server's documentation to determine its name. The return value from this function is an object reference to the Automation object, with which you can access properties and methods of the object.

For example, the following OpenScript statement calls the createAutoObject function, sends it the name of the Microsoft Word Automation server application as a parameter, and then stores the function's return value, which is a reference to the Automation object, in a variable named refWord.

```
refWord = createAutoObject("Word.application")
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

A connection to the Automation server is maintained as long as its object reference is stored in a variable. When there are no variables storing a reference to the Automation object, the connection to the Automation server application will be terminated. If you need to maintain a connection to an Automation server across multiple handlers or object scripts, you must store the Automation object reference in a system variable. It is important to remember to close the connection to the Automation object when you are finished working with it. Note that closing a connection to an Automation server does not close the server application. If you need the server application itself to shut down, you must call the appropriate method of the Automation object before you close its connection.

Writing OpenScript for ActiveX controls and Automation objects

ActiveX controls and Automation objects have properties, methods (similar to OpenScript functions), and events that you can affect using OpenScript code. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book. When writing OpenScript code for an ActiveX control or Automation object, note that Instructor automatically adds the prefix *ext* to the names of its methods, events, and properties to ensure that there are no conflicts with the OpenScript programming language. Use these Instructor-modified names when writing OpenScript code for an ActiveX control or Automation object.

Since many ActiveX controls are written for use in the Microsoft Visual Basic® programming environment, you'll want to understand how Visual Basic differs from OpenScript so that you can translate the control's documentation, which is likely to be written for Visual Basic developers, into its equivalent OpenScript code.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Suppose that you are working with an ActiveX calendar control named Cal1 and want to edit its caption. In Visual Basic, you would use the following command:

```
Cal1.Caption = "Calendar Control"
```

In OpenScript, the equivalent command would be:

```
extCaption of CalendarCtrl "Cal1" = "Calendar Control"
```

Note that the reference to the control in the Visual Basic example consists only of the object's name, *Cal1*. In OpenScript, the reference to the control must include the control's object type and its name or ID number, just as with any other Instructor object. In this example, the name of the control is *Cal1*, and the object type is *CalendarCtrl*.



Tip You can determine a control's type from the text displayed in the Instructor status bar when the pointer is paused over its tool on the tool palette, or over the control itself after you have drawn it on the page or background.

To determine which properties, methods and events a particular ActiveX control or Automation object supports, refer to its documentation.

Setting properties of ActiveX controls and Automation objects

Like any Instructor object, ActiveX controls and Automation objects have properties that you can set. ActiveX controls can have both persistent and nonpersistent properties. Persistent properties retain their value when the Instructor file is saved. Nonpersistent properties must be set at Reader level using action sequences or OpenScript. Automation object properties are always nonpersistent. To determine which properties a particular ActiveX control or Automation object supports, check the documentation for the ActiveX control or Automation server.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

The OpenScript code for manipulating properties of Automation objects is exactly the same as the code used to manipulate the properties of ActiveX controls. When working with ActiveX controls, it is common to store an object reference to the control in a variable, and then use the variable when accessing properties:

```
refCalendar = CalendarCtrl "Cal1"
extCaption of refCalendar = "Calendar Control"
```

Automation objects are very similar. Once the object reference has been created, you can access its properties and methods just as you do with an ActiveX control:

```
refWord = createAutoObject("Word.application")
refDocumentCollection = extDocuments of refWord
refMyDocument = extAdd("MyDocument.doc", 0) of extDocuments of \
   refWord
```

Most properties of ActiveX controls and Automation objects are simple values, but some properties are themselves objects, complete with their own properties and methods. For example, many controls also have a property called *font*, the value of which is also an object with its own properties. The following Visual Basic command sets the *bold* property of the *font* object of the calendar control to *true*:

```
Cal1.Font.Bold = true
```



Note ActiveX properties that have their own properties and methods are not supported in the Actions Editor, so you can only work with them using OpenScript.

When executed, this command causes the text displayed by the control to appear in bold typeface. The equivalent command in OpenScript would be:

```
extBold of extFont of CalendarCtrl "Cal1" = true
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Calling methods of ActiveX controls and Automation objects

In addition to properties, ActiveX controls and Automation objects have methods (similar to OpenScript functions) that you can call to control their appearance and behavior. Each ActiveX control and Automation object has its own, unique set of properties and methods. In order to determine which methods a particular ActiveX control or Automation object supports, check the documentation for the ActiveX control or Automation server.

For example, you can write a script that controls the behavior of Web Browser, the Microsoft Internet Explorer ActiveX control. Adding this ActiveX control to your application allows you to display Web pages in your book in response to a user action (in this case, clicking a button). When the user clicks the button, the following script will request a URL from the user and then tell the control to navigate to that URL. It does this by calling *extNavigate*, a method of the Web Browser control, and passing it a string containing the URL:

```
to handle buttonClick
   ask "Please enter the web address"
   destinationURL = it
   if destinationURL is not null
      get extNavigate(destinationURL) of WebBrowser "browser"
   end if
end buttonClick
```

In this example, the name of the control is *browser,* and the object type is *WebBrowser*. The method *extNavigate* is a method specific to the Web Browser control and can be identified as such by the prefix *ext*.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

A similar example uses Automation to display the list of templates currently defined in Word:

```
to handle buttonClick
   refWord = createAutoObject("Word.application")
   templateCollection = extTemplates() of refWord
   numTemplates = extCount of templateCollection
   message = "Word has " & numTemplates & " templates defined:" & CRLF
   step i from 1 to numTemplates
      templateFileName = extFullName of extItem(i) of \
         templateCollection
      put templateFileName & CRLF after message
   end step
   request message
   refWord = NULL
end
```

In this example, a new Word Automation object is created and stored in the *refWord* variable. Once the Automation object reference is obtained, the *extTemplates* method is called to return a collection of templates that have been defined for Word. The code loops through each item in the template collection and constructs a message displaying the filename of each template.

Handling Events from ActiveX controls and Automation objects

Like Instructor's own objects, ActiveX controls and Automation objects send messages, called events, which notify Instructor that something has happened to the control or object. Instructor can respond to these events, just as it does with regular ToolBook messages.

In order to determine which events a particular ActiveX control or Automation object supports, check the documentation for the ActiveX control or Automation server.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Handlers for ActiveX or Automation events are very similar to handlers for regular ToolBook events. In the case of ActiveX controls, the event message is sent directly to the ActiveX control object. Just as ActiveX property names and methods all start with *ext*, so do its events. The following example responds to the DownloadComplete event generated by Web Browser, Internet Explorer's ActiveX control:

```
to handle extDownloadComplete
   caption of statusBar = "The download is complete."
end
```

Responding to events generated by Automation objects works slightly differently. Since Automation objects are created through OpenScript and only exist as a reference stored in a variable, there is no actual object on a page or background to receive the event. When you create an Automation object using the createAutoObject() function, you can specify which ToolBook object will receive any events that are generated by it. To specify the target object for Automation events, pass an object reference of the object to be notified in the optional second parameter of the createAutoObject() function:

```
refWord = createAutoObject("adodb.connection", self)
```

This example creates an ActiveX Data Objects (ADO) connection object for accessing databases. Any events generated by the object will be sent to the object containing this script. As with ActiveX events, events generated by Automation objects will always be prefixed with *ext*.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Creating action sequences for ActiveX controls

In the Actions Editor, you can create an action sequence that responds to an ActiveX event. In turn, Instructor can manipulate the ActiveX control by changing its properties and executing its methods. In the Actions Editor, you can create an action sequence that sets the properties of a control or executes an ActiveX method.

Note You cannot create or control Automation objects using the Actions Editor.

Setting ActiveX properties

You can insert an action to change a property of an ActiveX control as part of any action sequence or shared action sequence. To set the property of an ActiveX control, you can use the Set Property action in the Actions Editor.

Note ActiveX properties that take certain data types and complex properties that have their own sub-properties are not supported in the Actions Editor. If necessary, you can manipulate all ActiveX properties using OpenScript.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To set an ActiveX control's property in an action sequence:**

1. Open the action sequence or shared action sequence in which you want to set a property of an ActiveX control.
2. From the Insert menu, point to Action, point to Object, and choose Set Property.

   The Properties for Action dialog box appears unless you changed this default setting.
3. In the Select the object for which to set a property list, select the ActiveX control.

   Only named controls will appear in the Select the object for which to set a property list. To name the control, right-click the control, and then choose Name to open the Name dialog box.
4. In the Select a property to set list, select the ActiveX control property you want to change.
5. Under Property value, select one of the two options, depending on whether you want to set the property to a literal value or the result of an expression.
6. Do one of the following:
   - Type in a literal value.
   - Type an expression.
   - Click the Build expression button and create an expression in the Build Expression dialog box.
7. Click OK.
8. In the Actions Editor, from the File menu, choose Update Actions & Close.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Executing ActiveX methods

An ActiveX method causes the ActiveX object to perform a specific task and often returns a value. Most ActiveX methods have parameters; when you execute an ActiveX method, you supply the value of those parameters. You can execute an ActiveX method as part of any action sequence or shared action sequence using the Execute ActiveX Method action.

➤ **To execute an ActiveX method in an action sequence:**

1. Open the action sequence or shared action sequence in which you want to execute a method of an ActiveX control.
2. From the Insert menu, point to Action, point to Object, and choose Execute ActiveX Method.

   The Properties for Action dialog box appears.
3. In the Select an ActiveX control list, select the ActiveX control whose method you want to execute; or, use the Choose object button to select the object. (The object must be named in order to select it.)
4. In the Select a method to execute list, select the ActiveX control's method you want to execute.
5. If the method requires parameters, click in the field under Parameter Value, and do one of the following:
   - Type in a literal value.
   - Type an expression.
   - Click the Build Expression button and create an expression in the Build Expression dialog box
6. *Optional.* In the Specify a variable to store a return value list, select a variable.
7. Click OK.
8. In the Actions Editor, from the File menu, choose Update Actions & Close.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Handling ActiveX events

Like Instructor's own objects, ActiveX controls send messages, called events, which notify Instructor that something has happened to the control. You can create an action sequence to respond to an ActiveX event.

➤ **To create an action sequence that responds to an ActiveX event:**

1. Select the ActiveX control.
2. From the Object menu, choose Actions.

   The Actions Editor appears.
3. In the Event list, select the ActiveX event.
4. In the Actions Editor, create an action or series of actions to respond to the ActiveX event you selected in the Event list.
5. In the Actions Editor, from the File menu, choose Update Actions & Close.

Troubleshooting ActiveX and Automation

If you have difficulty making an ActiveX control or Automation object work in Instructor, you may need to do some investigating to determine the cause of the problem. The first thing you should do when acquiring a new ActiveX control or Automation server is to read its documentation or release notes.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Troubleshooting ActiveX controls

The following tips may help you when working with ActiveX controls in Instructor:

- **Make sure that the control is installed and registered correctly** ActiveX controls are software components that must be installed correctly on your machine. Many controls will work only if you have their license file (a text file containing the license agreement) on your computer. If you find that a control does not work, try reinstalling it and make sure that you have the license file in the correct location on your machine.
- **Test the control in other environments** Since many controls are written to work in Microsoft Visual Basic, you may want to use it to test your control. Check the documentation or contact the vendor of the control to determine whether there are specific environments in which the control is intended to work.
- **Try the control in Microsoft Internet Explorer** Many controls that are written for the Internet depend on the Microsoft Internet Explorer browser for some of their functionality. If you find that an ActiveX control created for the Internet does not work as you expect within Instructor, try using the control in Internet Explorer.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Troubleshooting Automation objects

The following tips that may help you when working with Automation objects in Instructor:

- **Make sure that the Automation server application is installed and registered correctly** When you attempt to create an Automation object, Instructor asks Windows to locate the appropriate Automation server application. The server application must be installed properly on the computer to successfully create the object. If the server application is correctly installed and you are unable to successfully create the Automation object, check the Automation server name you are passing to the createAutoObject() function.
- **Test the Automation object in other environments** Since many Automation servers are written to work with Microsoft Visual Basic, you may want to use it to test your Automation scripts.

Additional Information

The discussion of working with ActiveX controls and Automation objects has been limited to the information you need to work with these technologies in the ToolBook II Instructor authoring environment. Although the method for accessing them is the same, each ActiveX control and Automation server will have its own set of properties, methods and events. Be sure to check the documentation for the ActiveX control or Automation server for more information.

If you would like more detailed information on these technologies, check out one of the many books on the market. Another valuable source of information is Microsoft's developer's Web site, *MSDN online*:

http://msdn.microsoft.com
http://msdn.microsoft.com/library

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

About ADO

ADO refers to Microsoft's ActiveX Data Objects, which is a high-level data access mechanism based on Automation. Instructor's Automation support gives it the ability to access a variety of databases through ADO, including Microsoft Access. As with any other Automation server application, ADO must be properly installed to access its features and functions. Information and documentation about the ADO object model can be obtained from a variety of sources.

To see an example of how ADO can be used with Instructor, explore the ToolBook II Instructor Quiz Builder (Samples\ADO\Quizbldr.tbk on your Instructor program disc). The Quiz Builder is a sample application that demonstrates one use of ADO technology—using ADO to send data to and retrieve data from a Microsoft Access database. For more information about the Quiz Builder and ADO, refer to the Quiz Builder online book (Samples\ADO\Quizbldr.pdf on your Instructor program disc).

About OLE

OLE is a technology based on a linking and embedding model that is used to integrate Windows programs. Using OLE, you can create an object in one application (the server application) and then incorporate it into another application (the client application). Because OLE objects retain information about where they were created, you can double-click an OLE object in the client application and edit it using the user interface of the server application. With OLE, you can create applications that integrate the capabilities of many different Windows programs. OLE objects are different from ActiveX controls in that an OLE object has a server application that provides its functionality.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

OLE objects can take as many forms as you have server applications: Worksheets or charts from a spreadsheet program (Microsoft Excel), word-processing documents (Microsoft Word), bitmaps and drawings (Microsoft Paint™), sound files, animations, digital video, and more. For example, you can create a worksheet in Excel (the server) and bring it into Instructor (the client) using OLE. You can then double-click the OLE object in Instructor and edit the worksheet in place. Similarly, you can embed a sound file as an OLE object in Instructor and then double-click the object to invoke Microsoft Sound Recorder to edit or play the sound clip.

An embedded OLE object is a copy of the object stored within your book, which makes it easy to move your application to another computer. Instructor stores information about which application created the embedded object so that you can invoke its server application if you want to edit or use the object. When you distribute an application that includes an OLE object, your users must have the OLE server application installed on their computer in order to edit the OLE object.

This version of Instructor supports OLE 2 technology but no longer supports the older OLE 1 technology. If you are working with a project that includes OLE 1 objects, you will need to replace them with OLE 2 objects. (OLE 1 objects will appear as rectangles in your application.)

Working with OLE

You can embed an OLE object in Instructor in two ways. One way is to create the object in the server application first and then add it to Instructor. Another way is to create an empty OLE object in Instructor and then use in-place editing or a copy-and-paste operation to import data from the server application.

When you embed an OLE object, the object retains the information about what application created it, but it does not keep a reference to the source file. Since there is no link, changes in the source file are not reflected in the embedded object in your book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Creating a new OLE object in Instructor

You create and embed an OLE object using the Insert Object dialog box, shown in Figure 2. This dialog box lists all the available server applications for objects that Instructor can embed. In this dialog box, you have the option to create a new, blank OLE object or to create an OLE object from a file that you have already prepared in a server application. When you create an OLE object, the OLE server application is added as an extension to your book.

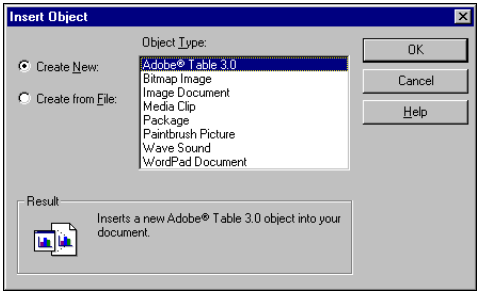


Figure 2
Insert Object dialog box

➤ **To create an OLE object in Instructor:**

Using the interface

1 Display the page or background in which you want to embed the object.

2 From the Insert menu, choose OLE Object.

The Insert Object dialog box appears.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

3 Do one of the following:

- To create a new, blank OLE object, choose Create new, and then select an object type in the Object type list.
- To create an OLE object from a file you've already prepared in the server application, choose Create from file, click Browse to locate and select the file on your computer, and then click OK.

4 In the Insert Object dialog box, click OK.

Instructor automatically creates an OLE object in your book and displays the object you've created. A tool appears on the Instructor tool palette that you can use to draw another OLE object in your book.

Using OpenScript

- To open the Insert Object dialog box, send the *insertOLEObjet* command:

```
send insertOLEObject
```

Editing an OLE object

You can edit an embedded object without leaving Instructor, a process known as in-place editing. When you double-click an OLE object in Instructor, the in-place editing capabilities of the object take effect, and a hybrid user interface appears. This hybrid interface combines tools from the server application with the client application, allowing you to edit the object without leaving Instructor.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To edit an OLE object:**

1 Double-click the OLE object.

The object is activated and a user interface from the server application appears in combination with the Instructor user interface.

2 Edit the object using the server application's interface.

3 When you have finished editing the object, double-click outside the object to close the server application's interface and return to Instructor.

Setting an OLE object's properties

OLE objects have properties that you can set using the Properties dialog box or the Property Browser.

➤ **To set the properties of an OLE object:**

Using the interface

1 Select the OLE object.

2 Do one of the following:

- Click the Properties button on the Instructor toolbar.
- From the View menu, point to Browsers, and then choose Property.

3 Set the properties you want for the object.

4 Close the Properties dialog box *(optional)* or Property Browser.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

- The *allowAuthorActivate* property determines whether you can edit the OLE object at Author level. If the *allowAuthorActivate* property is set to *true*, you can activate the OLE object by double-clicking it; if set to *false*, you cannot activate the object by double-clicking it.
- The *allowReaderActivate* property determines whether you can edit the object at Reader level. If the *allowReaderActivate* property is set to *true*, you can activate the OLE object by double-clicking it; if set to *false,* you cannot activate the object by double-clicking it.

Removing an OLE object

After you have added an OLE object to your book, it will remain an extension of your book until you remove it. Deleting an OLE object from a page does not remove the extension from your book.

➤ **To remove an OLE object from your book:**

1. Delete all instances of the OLE object from your book.
2. From the File menu, choose Extensions.
3. In the Extensions dialog box, select the OLE object you want to remove, and then click the Remove button to remove the selected extension from the Added extensions list.
4. Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Chapter 22

Optimizing Instructor applications

This chapter explains a number of techniques you can use to make your native ToolBook II Instructor and exported DHTML applications run as efficiently as possible. There are tips here for designing books, pages, and objects. This chapter also describes methods for benchmarking your OpenScript applications to determine how fast individual components run.

IN THIS CHAPTER

Tips for optimizing a book designed for DHTML export **610**

About optimizing OpenScript applications **611**

Optimizing books, pages, and objects **611**

Improving navigation and search speed **615**

Optimizing scripts **618**

Creating bitmap images of pages or backgrounds **625**

Benchmarking performance **630**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Tips for optimizing a book designed for DHTML export

Several factors affect how quickly an exported DHTML application displays in a Web browser. The speed of the Internet connection, the size of the media files in an application, and the complexity of the page design all contribute to the length of time necessary to download an application. You can minimize the amount of time it takes to run your application in a Web browser by building a book that is designed to use objects and media files efficiently.

➤ **To reduce the amount of time required to display your application in a Web browser:**

- Use one background in the book, or use the fewest number of backgrounds possible.
- Limit the number of objects on the page.
- Use graphics with a small file size. If bandwidth is an issue, reduce the overall size of your graphics.
- Use the Universal Media Player from the Catalog to show streaming media files. Due to the large file size of AVI files, you should not use the AVI file format for playing videos over the Internet.
- For question feedback, if you choose to play an audio file in response to a user's answer, use a short AU file or play a streaming media file in the Universal Media Player. You can make the Universal Media Player invisible if you choose.

While developing your book you should periodically export the book to DHTML and evaluate the result in a Web browser. Try to test your exported application using the same type of Internet connection as the intended audience.



Note Your book will go through the export process much faster if you use the Web Graphic Placeholder object to display GIF or JPEG graphic files on the pages of your book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

About optimizing OpenScript applications

Optimizing a ToolBook II Instructor application ensures that it runs as efficiently as possible, in terms of speed and the amount of memory it requires. Pages or individual objects that operate inefficiently can slow Instructor down when navigating to a page, running animation sequences, or scrolling through text. You should think about optimizing your application at all stages of its development: design, development, and testing.

The efficiency of some types of scripts can affect many aspects of your application, from how quickly a user can navigate in your application to how responsive the application is to keystrokes or mouse events. Much of this chapter describes how to optimize your OpenScript code. You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

Optimizing books, pages, and objects

Often, the goals of reducing memory usage and increasing display speed are complementary. For example, a smaller page both takes less memory and appears faster; the same is true for a page with few objects. In other cases, however, you may need to balance a number of factors. For example, if you store a bitmap image of a page, the page can appear faster, but the book requires more memory. In such cases, you need to determine which strategy suits your application best.

By observing the following guidelines for books, pages, backgrounds, and individual objects, you can optimize the speed and memory requirements of your application.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Making books more efficient

Your design of individual pages and objects can affect how quickly a book runs and how much memory it requires; these topics are addressed separately later.

➤ **To make a book more efficient:**

- Periodically compact the book by saving it under a different name with the Save As command or *saveAs* message. This process eliminates unused objects and saves memory.
- Use as few objects, pages, and backgrounds as possible, and delete unused objects to reduce memory use and speed display time.
- Make page size as small as possible so that the book displays the pages faster.
- Put as few statements as possible in the *enterApplication* or *enterBook* handler, and handle other setup tasks in the *enterBackground* handler while *sysLockScreen* or *lockScreen* is *true*. The fewer statements in an *enterApplication* or *enterBook* handler, the faster Instructor can run the handler and display the first page.
- Use *firstIdle* to execute a script after a page appears.
- Specify a book palette to minimize palette flash on 256-color display devices and to increase the speed with which your images appear.

Making backgrounds and pages more efficient

The size and number of pages and backgrounds in a book affect both the speed and the memory required to run your application.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To make pages and backgrounds more efficient:**

- Use only as many objects as you need. Fewer objects allow faster transition between pages. For details about improving speed by turning groups of objects into a bitmap, see "Using groups efficiently," later in this chapter.
- Use as few backgrounds as possible. If your application has several similar pages, design the pages so that they can share a common background.
- Set the *imageBuffers* property of viewers with complex images or backgrounds to *1* or *2*. Use two image buffers to improve navigation speed between pages that share the same background, or when the application moves or resizes objects over a complex background. Otherwise, use one image buffer to save the overhead of maintaining a second one. To save memory, set *imageBuffers* to *0* for pages used as dialog boxes and pop-up viewers. For details about image buffers, see Chapter 19, "Creating windows with viewers."
- Create an image of a page so that Instructor can display the page without having to compose it first. For details, see "Creating bitmap images of pages or backgrounds," later in this chapter.

Positioning objects for maximum efficiency

The speed of your application can be dramatically affected not only by how many objects there are on a page, but also by where you place the objects on the page or background.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To position objects for maximum efficiency:**

- Place objects that don't change (especially large paint objects and pictures) on the background, and keep objects that do change on the foreground. If objects on the background change while the application is running—for example, if they move, change size, or change layer order—Instructor slows down because it must redraw not only the background objects but foreground objects as well.
- Arrange foreground and background objects so that changes affect only small areas. This arrangement minimizes redrawing when a change occurs.
- Avoid overlapping objects, especially foreground and background objects. If such overlapping objects change, Instructor must redraw both the foreground and background areas to update the screen image. This guideline applies particularly to record fields; when another object overlaps a record field, Instructor must redraw both the record field and its text when navigating to that page.

Overlapped objects *Objects separated by at least 30 page units*

Figure 1
Overlapped objects slow display speed

Note An object has a one- or two-pixel "aura" that extends past its visible boundaries onscreen. To prevent overlap, leave at least 30 page units between objects.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Improving navigation and search speed

In addition to the guidelines listed in the preceding sections for designing pages and placing objects, use the following techniques for improving navigation and search speed.

➤ **To improve navigation and search speed:**

- Set *sysHistoryRecord* to *false* so that Instructor doesn't require time and memory to maintain *sysHistory* when you navigate from page to page. For details about *sysHistoryRecord*, refer to Chapter 8, "Writing common scripts," in the *Programming in OpenScript* online book.
- Set *imageBuffers* to *2* to increase the speed of navigation between pages that share the same background, especially if the background is complex.
- For short animation sequences that flip pages, preload the pages into memory so that when you run the animation sequence there is no delay while Instructor finds and loads the pages. Set *sysLockScreen* to *true*, flip all the pages, and then set *sysLockScreen* to *false* before showing the Instructor main window. (Use *lockScreen* if you are flipping pages in a viewer.)
- When searching, sorting, or navigating between pages, set *sysSuspendMessages* to *true*. This technique prevents Instructor from sending the *enterBackground*, *enterPage*, *enterField*, *enterRecordField*, or *enterButton* message each time it navigates to a new page and avoids the overhead of these messages.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

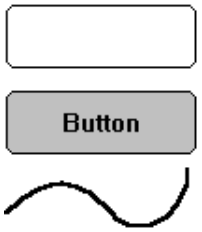
Index

Creating efficient graphics and draw objects

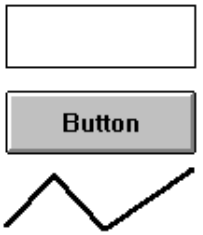
When you create an object or import a graphic, you can determine how quickly it appears by choosing particular object types or setting certain properties.

➤ **To create efficient graphics and draw objects:**

- Avoid objects with rounded corners, because Instructor displays these more slowly than other objects. Use *pushButton*, *rectangle*, and *shadowed* border styles to increase the speed with which buttons appear; avoid arcs, curves, ellipses, pies, and rounded rectangles.



Objects with rounded corners



Objects with square or angled corners

- Minimize the use of transparent objects (including bitmaps); they appear more slowly than other objects because Instructor must redraw both the object and the objects behind it whenever there is a change.
- If you are displaying a bitmap only once in your application, create it as a paint object rather than as a book resource. Graphics stored as resources in the book appear more slowly than those stored directly on the page. However, if you intend to use a bitmap more than once in your book, a bitmap resource uses less memory and storage space.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Making fields efficient

Take special considerations into account when working with objects that contain text, such as text fields and record fields.

➤ **To make text fields and record fields efficient:**

- Set *drawTextDirect* to *true* to display and scroll text faster, especially if the field displays a large quantity of text.
- Avoid using transparency in fields, and be certain that no other object overlaps a field. During page navigation, Instructor displays opaque fields and objects that don't overlap faster.
- To clear the text of fields in a group quickly, set the group's *text* property to *null*, which clears the text from all of the fields in the group at once.

Using groups efficiently

If you have created groups of objects in your application, observe all the guidelines given earlier about optimizing individual objects. In addition, follow the guidelines below, which are unique to groups.

➤ **To use groups efficiently:**

- Set the properties quickly for all objects in a group by setting group properties. Any properties that you set for the group are set for the individual members of the group (except for the *visible* property). The fastest method to clear the text of fields in a group is to set the group's *text* property to *null*, which clears the text from all of the fields in the group at once.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

- When displaying a complex group of objects, create a bitmap image of the group; Instructor may display a paint object (bitmap) faster than a complex group that is the same size. If you're using a group of objects to create an icon, for example, press ALT+PRINT SCREEN to copy the screen to the Clipboard, and then paste it into a Windows bitmap-editing program such as BitEdit (available from the Tools menu). Select the portion of the image you want to use, copy the selection to the Clipboard, and then paste it back into Instructor as a paint object by choosing Paste Special from the Edit menu.

Optimizing scripts

If you understand how Instructor runs OpenScript code, you can create scripts that run faster and consume less memory. For example, if you know how Instructor works with data types, you can use variables efficiently to store data.



Tip Users generally perceive applications to run faster if something happens onscreen while scripts execute. For example, you can display the hourglass cursor (set *sysCursor* to *4*), or you can use a viewer to display a progress indicator. For details about using viewers to display a progress indicator, see Chapter 19, "Creating windows with viewers."

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

As with books and pages, your goals are to achieve maximum speed and use minimum memory. The same technique often accomplishes both goals. For example, short strings are faster to process and use less memory. At other times, however, you must decide which is more important: speed or memory. For example, information in a system variable is faster to access than information in a user property. However, system variables require more memory and are valid only for a single instance of Instructor. You must choose which strategy is best for your situation.

An additional factor to consider when creating scripts is where to place them. There is very little overhead when messages pass up the object hierarchy, so you can take advantage of shared scripts and the hierarchy without a performance penalty. As a result, you can reduce scripting time by creating modular scripts that can be reused easily with little maintenance.

Writing efficient handlers

Some OpenScript statements and structures are less efficient than others. By minimizing the use of inefficient statements, you can avoid performance slowdowns in your application.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To write efficient handlers:**

- Avoid using the *execute* statement if possible, especially inside loops. Instructor recompiles the text following this command, a process that can affect performance substantially. Use *execute* only to create the needed statement when the application is running.
- Minimize the number of times you have to get the property of an object, because getting a property is slower than getting the value of a variable. For example, if you are extracting words individually from the text of a field, copy the text to a local variable and then work with the variable instead of the property:

```
--Copies text of one field to another,
--eliminating duplicate words
to handle buttonClick
   --Copies text to local variable
   fieldText = text of field "sample"
   newText = null
   step ctr from 1 to wordCount(fieldText)
      nextWord = word ctr of fieldText
      if nextWord is not in newText then
         put nextWord & space after newText
      end if
   end step
   --Sets text of new field object
   text of field "sample2" = newText
end buttonClick
```

Using variables effectively

You can improve the performance of scripts by using data types and arrays and by limiting the memory overhead of system variables. For details about data types and arrays, refer to Chapter 7, "Using variables, values, and text," in the *Programming in OpenScript* online book.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To use variables effectively:**

- Use data types as much as possible. Instructor can then store and manipulate the data in its most efficient format.
- When you use data types, choose one that is suitable to how you will use the variable. For example, the counter in a step loop works most efficiently if it is a *long* data type.
- If you don't explicitly assign a data type of a variable, use statements and functions that maintain a variable in a single format as much as possible. Instructor chooses a data type for the variable based on how the variable is being used in a script. For example, if you use the function *itemCount( )* with a variable, Instructor requires that the variable be a list (a *stack* type) and converts it to one if necessary. If Instructor must convert between formats, the script slows down.
- Use arrays to represent lists and matrices (tables). Accessing individual elements of an array is faster than accessing individual elements in a comma-delimited list.
- Limit the number of system variables you use, because they take up memory, especially if you put large quantities of information in them. As an alternative, you can put information into the text of hidden fields or into user properties, which are also global. The information is slower to extract but takes up less memory.

Using text efficiently in scripts

Working with text is a common requirement in scripts, so using it inefficiently can cause applications to slow down. It's especially important to choose the right operators, commands, and functions when working with text.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To use text efficiently in scripts:**

- Whenever possible, use *character* or *char* to refer to part of a string, because the character string specifier is much faster than *word* or *textline.* Specifying *word* is next fastest, followed by *item*, and then *textline*.
- Use quotation marks with any literal string so that Instructor doesn't have to determine first whether the string is a variable. For example:

```
get text of field Sample
get text of field "Sample"        --This statement is more efficient
```

- To find a character within a larger string, use the *offset( )* function rather than moving through the string character by character and comparing to the substring.
- When working with lists (the *stack* data type), use the *push* and *pop* operators to add and remove elements whenever possible. These commands are faster than using the *item* operator.
- To determine whether an item is in a list, use the *is in* operator, which is faster than popping the elements individually and comparing them to a value. However, be aware that *is in* searches for characters and not whole words (for example, it will find "smith" in a list containing "Smith," "Smithson," and "Goldsmith").
- Minimize the use of the *richText* property, because it requires more memory than ordinary text.
- Avoid reformatting text repeatedly, because doing so makes the RTF version of the text larger, requiring more time to update; the less reformatting, the shorter the RTF version of a field's text. To clear all formatting from a field, set its *richText* property to the *text* property:

```
richText of field "desc" = text of field "desc"
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Placing handlers for maximum efficiency

The Instructor object hierarchy gives you great flexibility in where you place handlers for messages. By taking advantage of the hierarchy, you can minimize the number of handlers that you have to create and maintain, making your application more modular.

➤ **To place handlers for efficiency:**

- Make one handler take the place of several similar ones by using a shared script. For example, if you're creating buttons and hotwords for navigation, you can use a single shared script to handle the navigation functions. For details, refer to Chapter 5, "Handling messages, events, objects, and properties," in the *Programming in OpenScript* online book.
- Instead of placing individual handlers in the script of each object, place a handler in the page, background, or book that contains the navigation statement. The handler can use the *object* or *name* and *target* properties to determine which object the user clicked. For example:

```
--Navigates to page matching the name of the button
--(a single handler works for a group of buttons)
to handle buttonClick
   conditions
      when name of target is "Next"
         --Use "send next" instead if you need
         --to forward the message
         go next page
      when name of target is "Previous"
         go previous page
      when name of target is "Beginning"
         go first page
         --Further statements here
   end conditions
   forward
end buttonClick
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

- Create composite objects such as toolbars or palettes by grouping objects and then putting handlers into the script of the group. This technique makes it easier to change the behavior of the group as a whole.
- Use system books to store scripts that can be used in many different applications, such as handlers for user-defined functions. However, check the performance of applications that rely on handlers in system books; using an excessive number of system books can result in performance degradation. For details, refer to Chapter 8, "Writing common scripts," in the *Programming in OpenScript* online book.
- Use notify handlers to create self-contained objects that can be copied to new books without changing any scripts. For details about notify handlers, refer to Chapter 6, "Creating an OpenScript statement," in the *Programming in OpenScript* online book.
- To avoid slowing keystroke entry, place any handlers for keyboard-event messages in the script of the target object or as low as possible in the object hierarchy. For example, if you have a handler for a keyboard-event message in the script of the page, all keystrokes entered for any object on that page must go through that handler.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Creating bitmap images of pages or backgrounds

When Instructor draws objects as offscreen images (their *drawDirect* property is *false*), you can improve display performance by creating and storing a compressed bitmap of a background's or page's offscreen image. The page can appear more quickly the next time the user navigates to it because Instructor displays the image without first composing it offscreen. This feature is useful if:

- pages in your book contain many objects or complex images that take a long time to display.
- objects do not change when users interact with them (for example, buttons do not invert when a user clicks them).
- objects on the page do not move (for example, there is no animation or scrolling).
- disk space for your book is not a critical factor.

Storing page images may not be of benefit if:

- pages have only a few objects on them.
- you're concerned about disk space requirements for the book.
- your application will be used on a different display device or a display device with a resolution different from the one on which you are developing it.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To store the current image:**

Using the interface

1 Navigate to the page or background for which you want to store an image.

2 From the Object menu, choose Properties for Page or Properties for Background.

3 On the Behavior tab, select Store an image of this page.

Instructor stores a bitmap image of the current page or background with the book.

4 *Optional*. Close the Properties dialog box.

When the Store an image of this page option is selected, Instructor saves an image of the page or background each time you navigate to a new page or background at Author level, but not when a user navigates at Reader level.

Using OpenScript

- Set the *storeImage* property of a page or a background to *true*:

```
storeImage of this background = true
```

- To get a list of devices for which the image has already been saved, get the value of the *storedImages* property:

```
get storedImages of this background
```



Note You should use the option to store an image only after you've finished designing your application. If you store an image of the page and then change the page, the stored image will be out of date.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Storing bitmap images for different display devices

A stored image in Instructor is device dependent. Instructor creates a new stored image if an image stored earlier was created on a display device that differed in:

- logical pixels per inch; most Windows-compatible display devices use 96 logical pixels per inch.
- color, measured in the number of colors that the device can display.
- font size, such as the difference between the small fonts setting and the large fonts setting in a display driver.

For example, Instructor might store different images for VGA and SuperVGA, because VGA uses 16 colors and some SuperVGA devices use 256.



Note Instructor does not store different images based on the size of a display device. For example, if a 1024x768 device and a 640x480 device use the same number of colors and logical pixels per inch, Instructor stores only one image.

To store images for more than one device type, run the book at Author level on each system and flip through all of the pages. Instructor creates new stored images for each page previously stored based on the characteristics of the new system. When a user runs the book later, Instructor automatically uses the appropriate stored image for the current system. A list of images stored for different resolutions appears in both the Properties for Page and Properties for Background dialog boxes.

Although the stored images are compressed, they will increase the book's file size, so you must determine the appropriate tradeoff between file size and performance.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Updating stored bitmap images

Changes the user of the application makes to objects on the page or background are not saved automatically when a stored bitmap image is used. Therefore, if you store a bitmap image, use one of the following techniques to ensure that the stored image remains up-to-date:

- Place all changeable objects on the foreground and store only the background's bitmap image.
- Set *drawDirect* to *true* for changeable objects and set it to *false* for all objects that do not change. Objects with *drawDirect* set to *true* are not stored with the bitmap image. Use this method if your application requires changeable objects on both the page and the background.

Removing stored images

If you are no longer using a stored image, remove it to save space in your book. For example, if you have created stored images for several different display types but are using the book on only one device, you can save space by removing the stored images you don't need. If you want to retain a stored image, you must remove all images first and then create a new stored image for the device type you are using.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

➤ **To remove a stored image:**

Using the interface

1. Navigate to the page or background for which you want to remove the stored image.
2. From the Object menu, choose Properties for Page or Properties for Background.
3. Clear the Store an image of this page check box.
4. *Optional.* Close the Properties dialog box.
5. Save the book with a new name so that Instructor will recover the space used by the image.

Using OpenScript

◆ Set the *storeImage* property of a *page* or *background* to *false*:

```
storeImage of this background = false
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

Benchmarking performance

To determine how fast your application is running, you need to run performance tests, or benchmarks. By building performance tests into your application as you develop it, you can decide which approach to an application-development task is the best.

To measure elapsed time for actions such as flipping pages, running animation, or performing calculations, use the *timeGetTime( )* function from Windows MMsystem.dll. The following handler includes additional statements useful for performing a single test because it links and unlinks the DLL each time it is run:

```
to handle <message>
   linkDLL "mmsystem.dll"
      DWORD timeGetTime( )
   end linkDLL
   startTime = timeGetTime( )
   --Further statements to test here
   endTime = timeGetTime( )
   --Display elapsed time
   request "This action took" && (endTime-startTime) && \
      "milliseconds."
end <message>
```



Tip If your application uses the ToolBook II run-time system book, you can use the ASYM_Ticks( ) function to link the DLL for you.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index

If you need to test an action in more than one location in a book, place the benchmarking scripts in the page, background, or book script. For example, you might place the following handlers in the book script to test page flipping speed:

```
to handle enterBook
   linkdll "mmsystem.dll"
      DWORD timeGetTime( )
   end linkDLL
   forward
   send leavePage
end enterBook

to handle leavePage
   system startTime
   startTime = timeGetTime( )      --Initializes startTime
   forward
end

to handle enterPage
   system startTime, endTime
   endTime = timeGetTime( )
   --Displays elapsed time
   request "It took" && (endTime-startTime) \
      && "milliseconds to display this page."
      --Multiply by 1000 to get seconds
   clear startTime
   clear endTime
   forward
end
```

After you have finished benchmarking a process, you should remove the timing tests to eliminate needless performance overhead.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

17 Using resources

18 Importing and exporting data

19 Creating windows with viewers

20 Creating and modifying menu bars

21 Using ActiveX, Automation, and other Windows technologies

22 Optimizing Instructor applications

PART VI
Distributing and printing your application

Glossary

Index



PART VI

Distributing and printing your application

In this part of the *ToolBook II Instructor User Guide,* you will learn about the powerful distribution options available with ToolBook II Instructor. These chapters discuss how to prepare your application for release and explore the special issues involved when you choose to distribute your application on the Internet. This part also contains a chapter describing how to print from Instructor.

IN THIS PART

Chapter 23 Preparing applications for release

Chapter 24 Distributing applications on the Internet

Chapter 25 Printing from Instructor



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Chapter 23

Preparing applications for release

When you have finished authoring your application, organize and check your files and then prepare them for distribution. You can make your application available to users via the Internet or an intranet, a local area network, or a CD-ROM. This chapter describes how to check and test your application and how to prepare your files for distribution.

IN THIS CHAPTER

Organizing and checking your application **636**

Preparing your application based on a distribution method **642**

Using the AutoPackager **647**

Using Instructor on a network **652**

Installing and testing your application **653**

Uninstalling an application **654**



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Organizing and checking your application

When you have finished developing your ToolBook II Instructor application, a few additional tasks remain that will improve the performance of your application when it is distributed to users. To optimize your application's appearance and behavior, you can:

- organize the directory structure.
- check the spelling.
- remove unused resources.
- compress the books to their smallest size.
- reset the books so that the user can run them.

Creating a central book directory structure

It's a good idea to create an organized directory structure while authoring, and it's especially important to have this structure in place before you prepare your application for distribution. Take some time to make sure that all the files that users will need to run your application are in one place and that like items are grouped together.



Note If you plan to deliver your application locally (that is, if users will install your application on their computers), make sure that the directory structure you create appears the way you want it to appear on the users' systems.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

All of the books that make up your application should reside in one central book directory, with one or more subdirectories for the media (sound, graphics, video, and animation) files that accompany those books.

Let's say your application has just one book—Introduction to French. We suggest the following directory structure:

```
\French
   IntroFrench.tbk (the ToolBook II book)
   \French media
      EiffelTower.avi (a video file)
```

Now let's say your application is more complex. You have one main book about Romance languages and several related books about French, Spanish, Italian, and so on. You could organize the files using several book subdirectories under a central directory, as shown in this example:

```
\Romance
   Romance.tbk
   \Romance and shared media
   \French
      IntroFrench.tbk
      AdvFrench.tbk
      \French media
         EiffelTower.avi
   \Spanish
      IntroSpan.tbk
      AdvSpan.tbk
      \Spanish media
         PradaMuseum.avi
etc.
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Checking spelling

You can use the Instructor spelling checker to ensure that your application is free from spelling errors. You can also set the spelling checker to use a custom dictionary that you create or a foreign-language dictionary that you purchase.

➤ **To check the spelling in an application:**

1 From the Tools menu, choose Spelling.

The Spelling dialog box appears.

2 *Optional.* Do one or both of the following:

- Click the Options tab, and then select options to define the conditions of the spell check.
- Click the Dictionary tab to select the dictionary to use.

3 Click the Spell tab, and then click Start.

The Start button becomes the View button. When the spelling checker finds an unknown word or other error (such as a duplicate word), it highlights the word and displays the text in the Old box.

4 Use the buttons on the Spell tab to ignore or change the unknown word, to add the word to a user dictionary, or to skip to the next object or page. To view a list of alternative words, click Suggest. To have the list of alternatives appear automatically, select Automatic.

5 Continue checking words until the spell check is complete, and then click Close to exit the Spelling dialog box.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Checking for unused resources

Resources added to your book during development increase the overall file size of your project. If you have added bitmaps, fonts, menus, or other resources to your book that you did not use in any part of your application, you can remove those resources from your book to reduce the file size. For more information about resources, see Chapter 17, "Using resources."

➤ **To remove resources from a book:**

1. From the Object menu, choose Resources.
2. In the Available resources list, select the type of resource you want to remove.
3. In the graphical list of resources, select the resource you want to remove.
4. Click Remove to remove the resource.
5. Repeat steps 2, 3, and 4 until you have removed all unused resources from your book.
6. Click Close.



Important You cannot remove a resource from the book if it has been assigned to an object. The Resource Manager dialog box maintains a reference count, which reflects the number of times a resource is used in a book. This reference count is not updated for references to a resource within a script. Be careful not to remove any resources that are referenced in a script.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Checking for unused extensions

If you added an extension during development that you do not use in your book, it is still registered as an extension that your book needs to run. You should remove any extensions that are not used in your book before you package the book for distribution.

➤ **To remove unused extensions from a book:**

1 From the File menu, choose Extensions.

The Extensions dialog box appears.

2 In the Added Extensions list, select the unused extension.

3 Confirm that the extension is unused by looking at the number next to References. If the number is 0 (zero) the extension is not used in the book.



4 Click the Remove button to remove the unused extension from the Added Extensions list.

5 Repeat steps 2,3, and 4 until you have removed all unused extensions.

6 Click OK. (If you are prompted to add a selected extension to the Added Extensions list, click NO to avoid adding any additional extensions to your book.)

Compressing your book

While you are developing an application, the file size grows as you make changes and add content to your book. You can reduce the size of your file by saving it under a new file name. To ensure that Instructor compresses your book to its most efficient size, save it under a new file name and then save it again under its original name.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

We recommend that you save your file under a new name periodically while you are developing your application. One easy way to do this is to save a backup copy of your book under a slightly different file name. For example, if you are creating a book called Sample.tbk, create a directory called Backup; periodically choose Save As from the File menu and then save your book as BackupSample.tbk in the Backup directory. Immediately after saving BackupSample.tbk, choose Save As from the File menu and this time save your book as Sample.tbk in the original book directory. This simple task can greatly reduce your file size and has the added benefit of creating a backup copy of your book.

Regardless of whether you've been saving your book under a new file name periodically, you'll want to do this before you prepare your application for release. Be certain that after you save your book under the new file name, you save it once again using the original file name.

Resetting your book

When you choose Save or Save As from the File menu, you may have the option to reset your book. Resetting prepares your application to be run by the user by putting all objects and pages in their reset state. When you have finished developing your application, reset the book so that features like quizzes and animations are ready to run.

If you don't see the option to reset your book when you save, the option to display this dialog box may be turned off in the Properties for Book dialog box, or your book may not contain any objects that require resetting.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

➤ **To turn on the prompt for reset when saving:**

1 From the Object menu, choose Properties for Book.

2 On the General tab, select the Prompt for reset when saving check box.

3 *Optional.* Close the Properties for Book dialog box.

Preparing your application based on a distribution method

You'll prepare your application for release according to how you intend to distribute it. For example, if you expect users to view your application over the Internet, you'll use the ToolBook II Web Specialist to either export your application to DHTML or to prepare your file to be displayed in a Web browser using the ToolBook II Neuron plug-in. If you plan to have your users install your application on their computers, you'll gather the necessary files and create a setup program using the ToolBook II AutoPackager.

The remainder of this section describes how to prepare your files for the distribution method you've chosen. A summary of distribution methods and preparation steps is provided in the table on the following page.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Prepare your files for release based on a method of distribution

If you plan to deliver your application this way...	Do the following to prepare for release...
Via the Internet or intranet as a series of Web pages that utilize DHTML	Use the ToolBook II Web Specialist to export your application. Ensure that any graphics or ancillary files are compatible with Internet standards.
Via the Internet or intranet as a native ToolBook II application displayed in a Web browser using Neuron	Use the ToolBook II Web Specialist to prepare your application. Ensure that users have (or can download) Neuron and are running the Windows operating system.*
Via a local area network as a native ToolBook II application	If users will install the application on their own computers, use the ToolBook II AutoPackager to create a setup program. Ensure that users are running the Windows operating system.*
Via CD-ROM	Use the ToolBook II AutoPackager to create a setup program. Ensure that users are running the Windows operating system.*

* Users must have Windows 95 SR2 or later.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Preparing files for the Internet

You prepare files for distribution on the Internet using the ToolBook II Web Specialist. The Web Specialist is a wizard that steps you through the process of preparing your application for the Web, either as Web pages or as a native ToolBook II application that uses the ToolBook II Neuron browser plug-in to display your application in a Web browser. The Web Specialist gathers information about your application and then creates the files that you'll distribute to your learners. The Web Specialist can even copy your files to a Web server.

More information about using the Web Specialist is available in Chapter 24, "Distributing applications on the Internet."

Preparing files for a local area network

You prepare files for distribution on a local area network using the ToolBook II AutoPackager. Before using the AutoPackager, you'll need to decide whether users will install the course on their local computers by running a setup program from a central server, or whether they will run the course from the central server and view it on their local computers.

If you plan to have users install the course on their local computers, you should prepare your application for the AutoPackager using the same optimization methods you would if you were distributing via CD-ROM (see the following section). However, when you use the AutoPackager, select the local area network (LAN) distribution option.

If you plan to have your users run the course from a central server, simply organize your application and any media or ancillary files in an orderly directory structure. If you reorganize your directory structure during the development process, you can use the AutoPackager to update any media paths in your application. For more information about using the AutoPackager, see "Using the AutoPackager," later in this chapter.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Preparing files for CD-ROM

You prepare files for distribution on CD-ROM using the ToolBook II AutoPackager. Before using the AutoPackager, you can optimize the performance of your application by saving each book in the most efficient format for use by CD-ROM drives. When you choose to optimize for CD-ROM, Instructor saves related data—such as a page and its objects, graphics, and resources—in close proximity on the CD-ROM to enable faster access. Keep in mind that an application can be composed of multiple books. To optimize an application for CD-ROM, you optimize each book separately.

Optimizing a book for CD-ROM takes more time than a normal save (up to five times longer). In addition, if you make changes to the book after optimization, you must optimize the book again. Therefore, make optimizing for CD-ROM one of the last steps in preparing your application for the AutoPackager.

➤ **To optimize a book for CD-ROM:**

1. Save the book under a new name, as described in "Compressing your book," earlier in this chapter.
2. From the File menu, choose Save As.
3. Enter or navigate to the book's original file name.
4. Select Optimize for CD-ROM, and then click OK.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Using a cache file for a CD-ROM application

When you optimize a book for CD-ROM, you can specify whether the book uses a temporary or permanent cache file. A cache file, which is stored on the user's hard disk, contains information that Instructor needs to access frequently. Using a cache file can make a CD-ROM application run significantly faster, especially for users who have a single- or double-speed CD-ROM drive; however, cache files do not improve the speed of newer, faster CD-ROM drives. Cache files are used only when books are run from the CD-ROM, not when they are installed from the CD-ROM and run from the hard drive.



Important If you make changes to your book after creating a cache file, you must recreate the cache file.

➤ **To build a cache file for a book:**

1. Save the book under a new name, as described in "Compressing your book," earlier in this chapter.
2. From the File menu, choose Save As.
3. Enter or navigate to the book's original file name.
4. Select Optimize for CD-ROM.
5. Click Cache File Options to open the Cache File Options dialog box.
6. Under Build cache file, choose Temporary or Permanent. (If you choose Never, no cache file is created.) To significantly improve startup speed in a large book, use the Permanent option.
7. Under Cache file type, choose Minimal, Preferred, or Extended.
8. *Optional.* To preview the amount of disk space required for the cache file, click Show Sizes. (Previewing can take a significant amount of time.)
9. Click OK to close all dialog boxes.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Using the AutoPackager

The AutoPackager gathers all the files used in your application, records information about your application, and copies these files to one location. Use the AutoPackager if you plan to distribute your application on CD-ROM or on a local area network from which users will install the application locally.

You can also use the AutoPackager to update media paths if, during the development process, you moved your application files to a new location on your hard drive or to another computer.

The package created when you run the AutoPackager consists of all the files belonging to the application (such as book files, media files, extension files, and run-time files) as well as files that are required to install the application (archive files, setup program files, and related data files). When you package an application, the AutoPackager:

- gathers the media and application files needed to run your application and checks the path information for the media in your application.
- prepares the files necessary to install your application using the options you specified and creates the Setup and Package directories.
- optionally copies the package to a chosen destination to allow for testing the setup program and creating the CD-ROM.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

➤ **To package your book using the AutoPackager:**

1 From the File menu, choose AutoPackager.

2 Follow the instructions in the AutoPackager. The AutoPackager will:

- allow you to specify a name for your application.
- allow you to specify a default installation directory.
- check for required media files and extension files.
- update all media paths.
- give you the option to include ToolBook II run-time files with your application.
- allow you to specify which of your application's files are copied to the user's computer.
- allow you to disable the custom installation option and install only those files that you specify.
- give you the option to create Windows shortcuts for your application.
- create a package definition file (with an .asu file extension).
- create a setup program that learners can use to install your application.
- copy your application to a specified location.

During packaging, the AutoPackager records the actions it performs and any error or warning conditions it encounters in a log file; this file, named Packager.log, is located in the Package directory.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Gathering files

The AutoPackager gathers the media files associated with your application and updates the path information referenced in the media links of your book or books. You can choose whether the package includes the current (open) book or all of the files in the current book's directory and its subdirectories. For example, if you package an application that consists of several books and these books are located within the central book directory structure, each book is searched for references to media files. When the AutoPackager encounters a media file located outside of the central book directory structure, it can copy or move the file to the book's directory or a subdirectory of that book's directory. For more information about organizing book files, see "Creating a central book directory structure," earlier in this chapter.

Packaging files

Using the AutoPackager, you can specify the default installation options that determine whether media, ToolBook II, and cache files will be copied to the user's computer or whether they will remain on the CD-ROM.

When you package a book, the AutoPackager creates a Package directory and a Setup directory. The Package directory contains files that the AutoPackager uses to create the Setup directory. If you package the same book again, the AutoPackager can reuse the files in the Package directory and will process your book quickly. After you have finished packaging your book, you no longer need the Package directory.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

The Setup directory contains all the files required to install your application on another computer. In the Setup directory, you will find:

- the setup file, Setup.exe.
- the initialization file, Setup.ini.
- the description file, <Book name>.asu.
- the packaging list file, Setup.pkg.
- the TBSystem subdirectory (the run-time files).
- the <Project name> subdirectory (your book and media files in a media subdirectory, if you've created one).



Note A Setup directory is created only when you choose to use the AutoPackager's own capabilities for creating a setup program for your users. If you use InstallShield Express to create a setup program, the AutoPackager does not create this directory. For more information about using InstallShield Express, see "Creating a setup program using InstallShield Express," later in this chapter.

Copying files

When you create a package for distribution via CD-ROM, the AutoPackager copies the archive and setup files to the directory of your choice and creates any subdirectories within that directory. You can then test the setup program from this directory structure before you generate a CD-ROM.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Creating a setup program using InstallShield Express

Included on the Instructor program disc is InstallShield Express, a tool that allows you to create a setup program for your users. If you install InstallShield Express, the AutoPackager will give you the option to use either InstallShield Express or the AutoPackager's own capabilities for creating setup programs. Using InstallShield Express allows you greater flexibility in customizing installation options. For example, you can change the appearance of the setup program, add billboards that appear as your application is installed, display a custom license agreement, and much more.

If you choose to use InstallShield Express, the AutoPackager creates a project file that you can open and edit in InstallShield Express. This file will step you through the process of creating a setup program. InstallShield Express contains an online tutorial and an online Help system.

Your ToolBook II license agreement allows you to freely distribute courses using InstallShield Express.

Including the run-time version of ToolBook II

When you distribute your application, you need to include a run-time version of ToolBook II so that users of your application who do not have access to a full version of ToolBook II can run your application. The run-time version of ToolBook II is contained in the TBSystem subdirectory of the Setup directory created by the AutoPackager. A run-time version of ToolBook II allows users to run books at Reader level only.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Using Instructor on a network

You can store your Instructor application on a network to allow multiple users to access and work in the application. Instructor ensures that, under normal circumstances, books on network file servers are not corrupted by multiuser access.

There are two ways to a maintain network version of an Instructor application. You can:

- provide a read-only version of a book that can be opened and used by anyone. Users can open and copy a read-only book even if it is already open on the network, but they cannot delete the book, change its file attributes, rename it, or save changes. Other applications, including DOS utilities and Windows applications, can open the file and access information, but they cannot change its file attributes, rename the file, or save changes.
- provide a standard read/write version of a book that can be opened by only one user at a time. Users cannot open a read/write book, copy it, delete it, or change its file attributes until the first user has closed it. By default, users have read/write access to Instructor files.

➤ **To allow read-only access to an Instructor book:**

1. Right-click the Instructor file in Windows Explorer.
2. From the right-click menu, choose Properties.
3. On the General tab, select Read-only.
4. Click OK.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Installing and testing your application

If you are distributing your application via the Internet or an intranet, test your application for any unforeseen configuration issues by using the versions of Web browsers your users are likely to use. Particular points to check include:

- **Media** Make sure that the application runs in a Web browser and that your media clips play correctly.
- **Hyperlinks** Make sure that the hyperlinks to the pages within your application and the hyperlinks to URLs work. Be sure to test image maps and the jumps on your map or contents pages.
- **Page navigation** Make sure that the navigation buttons in your application work. To avoid losing your users in hyperspace, make sure that page navigation is clear and that you have provided ways for users to return to the pages they were viewing.

If you have created distribution CDs or a network directory containing setup files, install and test your application. Test on a system that does not have Instructor or ToolBook II Assistant (or any other version of ToolBook) installed to ensure that the application installs correctly and that all media files play.

➤ **To test your application's installation program:**

1. Insert the CD-ROM into your computer.
2. From the Start menu, choose Run.
3. In the Run dialog box, type *d:\Setup.exe*. The drive letter may differ; use the appropriate letter for your CD-ROM drive.
4. Click OK, and then follow the instructions on the screen.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Uninstalling an application

The package you create for distribution via CD-ROM includes the InstallShield Uninstall program, which removes the application and its files from the system.



Note When a user uninstalls an application, only files installed as part of the application are removed. Files created by the application remain on the hard drive.

When your users install your application, the Setup program adds entries to the Windows Registry, which then allows files to be automatically removed using the Windows Add/Remove Programs utility. This utility is located in the Windows Control Panel. Your application will be listed by the application name you assigned it in the AutoPackager.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Chapter 24

Distributing applications on the Internet

ToolBook II Instructor makes it easy to distribute your online learning application over the Internet. You can create a native ToolBook II application that can be displayed in a Web browser using the ToolBook II Neuron browser plug-in, or you can export your book as a series of Web pages that utilize DHTML technology. This chapter discusses the two methods for delivering your course on the Internet and outlines the steps you can take to prepare your application for use with a learning management system.

IN THIS CHAPTER

About Internet distribution options **656**

Choosing an Internet distribution option **657**

Publishing your book to the Web **662**

Using ToolBook II courses with a learning management system **667**

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

About Internet distribution options

Instructor offers two choices for viewing an application in a Web browser and allows you to choose the method that best meets your needs. You can export your application as a series of Web pages, or create a native ToolBook II application that can be displayed in a Web browser using the ToolBook II Neuron browser plug-in. In either case, you publish your book for use on the Internet with the ToolBook II Web Specialist, a wizard that gathers information about your requirements and steps you through the process of preparing your application for use on the Internet.

Exporting your application as a series of Web pages that incorporate DHTML allows learners with access to the Internet to view your lesson in a Web browser without the Neuron browser plug-in. While using DHTML allows for maximum interactivity—including behavior programmed using the Actions Editor—older versions of Web browsers do not support DHTML.



Note Scripts created using the OpenScript programming language do not export to DHTML. If you've added functionality to your application using OpenScript, you can deliver it over the Internet as a native ToolBook II application using the ToolBook II Neuron browser plug-in.

When you create a native ToolBook II application that your users view in a Web browser with the Neuron browser plug-in, you can use the complete set of features available in Instructor. Native ToolBook II applications can contain interactive custom features, including those created using OpenScript. Your users download and install the Neuron browser plug-in, and your applications run in a Web browser exactly as they run on a local computer.

How to use this online book

Contents

PART I Welcome to ToolBook II Instructor

PART II Building your application

PART III Adding interactive features to your application

PART IV Adding behavior with action sequences

PART V Using advanced features

PART VI Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Choosing an Internet distribution option

The distribution method you choose should reflect the requirements of both the developer and the user. As the developer, you will want to consider the following factors:

- **Interactivity** Instructor includes the option to automatically export your book to DHTML, an advanced Internet technology that allows authors to set actions for buttons, responses to questions, trigger prompts, use conditions and loops, and more.
- **Browser technology** Older versions of Web browsers (earlier than 4.0) will not support DHTML. Determine which browser versions your learners are likely to use, and then choose a distribution method that will take advantage of the technology.
- **Required component** The Neuron distribution method requires that your learners install the Neuron browser plug-in.

The table on the following pages summarizes the advantages and limitations of each Internet distribution method. The remainder of this section explores each distribution method in detail.



Tip You can change the distribution method for a book at any time. To select a different method, from the Object menu select Properties for Book and then click the Deployment tab.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Factors to consider when choosing an Internet distribution method

Distribution method	Advantages	Limitations	Browser requirements
DHTML	Using DHTML allows you to use a wide variety of interactive features, including actions programmed using the Actions Editor.	Older browser versions do not support DHTML.	Microsoft Internet Explorer 4.01 SP1 or later; Netscape Navigator 4.06 or later.
Neuron	Using the Neuron plug-in allows you to run a full-featured Instructor application in a Web browser, and can include OpenScript code.	Neuron is available only for the Windows platform. Users must install Neuron on their computers.	Internet Explorer 4.0 or later; Navigator 4.0 or later.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Understanding DHTML

HTML is a markup language that allows you to format documents for the Web. An HTML document combines your content with HTML markup codes, called tags, that specify how your content will be formatted. When you view an HTML document in a browser, you do not see the tags; you see only the formatting that the tags represent.

Dynamic HTML, or DHTML, extends HTML by adding the ability to create interactive features such as buttons that respond to a click, scrolling text, animations and more. The preprogrammed objects in the Catalog have these features built in. You can use the ToolBook II Web Specialist to automatically convert your application to Web pages that utilize DHTML. Exporting to DHTML from Instructor creates Web pages that are customized for Microsoft Internet Explorer and Netscape Navigator.

When you export to DHTML, most objects and features will appear and behave as they do in Instructor. The following table describes the functionality you should expect when exporting your application to DHTML.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Functionality to expect when exporting your book to DHTML

Object or feature type	DHTML
Action sequences	Objects programmed using the Actions Editor will behave as they do in Instructor.
Buttons	Button captions will not appear transparent unless the caption is centered. Access keys on button captions will not function.
Hyperlinks	Transition effects between hyperlinks will appear in Microsoft Internet Explorer, but not in Netscape Navigator.
Menu bars and viewers	These features will not function in an exported application.
OpenScript	Objects whose functionality has been extended using OpenScript will not function as scripted. Use the Actions Editor to program behavior that will export to DHTML.
Paint objects	If you have cropped a paint object in Instructor, the paint object will appear compressed when you view it in a browser. To avoid this effect, crop graphics in a graphics editing program before importing them into Instructor.
Question objects	Question objects will behave as they do in Instructor.
Text fields	Single-select list boxes, multi-select list boxes, and editable fields will lose their character formatting and will not appear transparent.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Understanding Neuron

Neuron is a browser plug-in and ActiveX browser control that enables a Web browser to play a ToolBook II book (a TBK file). Using Neuron, you can take advantage of the full feature set available in native ToolBook TBK files, including code written in OpenScript, and distribute an application over the Internet. Neuron is available for both Microsoft Internet Explorer and Netscape Navigator running on the Windows platform.

Neuron can also help you to maintain security while running an application over the Internet. Because ToolBook II applications are executable files that can fully interact with your computer, you should exercise some caution when accessing files from an unfamiliar source. To allow you to control whether a ToolBook II application can interact with your computer, Neuron has two modes of access: secure and nonsecure. Use nonsecure mode when accessing ToolBook II applications you are confident were created by reputable developers; use secure mode when accessing unfamiliar applications. When running in secure mode, Neuron will not allow links to system files and will not execute commands such as Save, Delete, or other operations that may affect your computer system.



Note Question objects will function only in nonsecure mode.

Impulse for Neuron is a utility that allows you to optimize your ToolBook II applications for distribution on the Internet. Impulse cuts the data in your book into pieces that can be reassembled by Neuron. This technique allows a book to be downloaded piece by piece on demand, providing increased speed performance over the Internet. This process is similar to the streaming technology used for transmitting audio and video files over the Internet.

ToolBook II Neuron and Impulse are available on your Instructor program disc. Neuron can be downloaded free from the click2learn Web site at www.click2learn.com.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Understanding ActiveX

ActiveX controls are programmed objects that perform specific tasks within a Web page or in native ToolBook applications. ActiveX controls allow you to add custom features to your application. An ActiveX control can be simple (such as a calendar) or complex (such as the Microsoft Web Browser control). ActiveX features are available only on the Windows platform.

For more information about using ActiveX, see Chapter 21, "Using ActiveX, Automation, and other Windows technologies."

Publishing your book to the Web

Whether you decide to deliver your book in native ToolBook II format or as a series of Web pages using DHTML, it's easy to publish your book to the Web using the ToolBook II Web Specialist. The Web Specialist is a wizard that gathers information from you about your application. By answering the questions in the Web Specialist, you are providing the information that Instructor needs to export your book.

You can use the file transfer feature in the Web Specialist to copy your files to a Web server. If you choose not to copy your files to a Web server from the Web Specialist, you can do this at a later time by opening the Web Specialist, clicking the Expert button, and selecting the option labeled Transfer your application to a server.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index



Figure 1
The ToolBook II Web Specialist

Tip Clicking the Expert button on the first page of the Web Specialist allows you to set export options in a streamlined dialog box. Use this format if you are comfortable with the export options and do not require the explanatory text that you'll find in the wizard format. You have access to all export options in either format.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Exporting to DHTML

You can use the Web Specialist to export your application to a series of Web pages that you can then choose to place on a Web server.



Note The Web Specialist exports your files to a project directory that you specify. If your application contains hyperlinks to other books, export those books to the same project directory.

➤ **To export your application as Web pages:**

1. Open the book that you want to export.
2. From the File menu, choose Publish to Web (DHTML).

   The Web Specialist appears.
3. Follow the instructions in the Web Specialist.

   Instructor exports your files using the settings you specified.

After you have exported your book to the Web, you can view it in a Web browser. When you export your book to DHTML, you choose a project directory. By default, Instructor names this directory WebExport and places it in the book's directory. Within this directory, Instructor creates another directory with the same name as your book file. There you will find all the files required to run your application on the Web. To start your application, open the index.html file.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Preparing a native ToolBook II book for the Web

You can use the Web Specialist to prepare a native ToolBook II book (TBK file) that your users will view in a Web browser using the Neuron browser plug-in.

➤ **To export your native ToolBook II book for the Web:**

1. Open the book that you want to export.
2. From the File menu, choose Publish to Web (Neuron).

   The Web Specialist appears.
3. Follow the instructions in the Web Specialist.

   Instructor exports your files using the settings you specify.

When you use the Web Specialist to prepare a native ToolBook II book, Instructor creates a list file that itemizes all of the files used in your application and notes their location relative to your book. Instructor saves the list file in the project directory that contains your book. This list file has the same name as your book but with an .lst extension. The Web Specialist uses the list file to copy all of the files in your application to your Web server.



Note It is important that you use the file transfer feature in the Web Specialist to copy a native ToolBook II application to a Web server. The FTP feature in the Web Specialist uses the list file to locate all the media files used in your application and copy them to the Web server while maintaining any relative paths.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

If you choose in the Web Specialist to use Impulse for Neuron to optimize the performance of your book over the Internet, Instructor creates the Impulse directory, in which it places the optimized file segments. These file segments have the same name as your project file, but each segment has a different extension. The first file segment has a .tbk file extension, and each subsequent segment has an incremented numerical extension (.001, .002, and so forth).

To run an application in a Web browser using the Neuron plug-in, you must create an HTML page that contains a link to your native TBK file. (To run an application that has been optimized using Impulse, create an HTML page with a link to the file segment with the .tbk extension.) You can create an HTML page using any HTML editor.

➤ **To create an HTML page that will launch a Neuron book:**

◆ In an HTML editor, create an HTML page with the following tag:

<a href="http://www.your_site.com/books/book1.tbk" TYPE = "application/toolbook">

where *www.your_site.com/books* is the location on your Web server where the native TBK file can be found and *book1.tbk* is the name of the native TBK file.

After you have created your HTML page, you can view it in a Web browser and test the link to your native TBK file.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Using ToolBook II courses with a learning management system

You can deliver courses created in ToolBook II over the Internet and use any AICC-compliant learning management system to record the score from student responses to questions. By adding evaluation buttons from the Catalog to your book, you allow results to be sent to a database that is controlled by the learning management system.

Adding score tracking to a ToolBook II course

When you use the Web Specialist to prepare your book for the Web, AICC support is added automatically to your book. This allows scoring information such as quiz results to be tracked and stored in a database. By including one or more buttons from the Scoring and Tracking category of the Catalog in your course, individual test scores can be sent to the learning management system in use. After a student opens a course through the learning management system and answers the questions in the course, the final score is available through a progress report generated by the learning management system.

A ToolBook II course can be stopped before completion and resumed at a later time without disrupting the overall score. You need to include a Suspend Lesson button from the Scoring and Tracking category of the Catalog in your course to allow a student to save the current state of the course before completion.

If a student accesses a ToolBook II course through a learning management system and closes it without selecting one of the exit buttons from the Scoring and Tracking category of the Catalog, no work accomplished during that session will be recorded.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Learning management solutions from click2learn.com

Two learning management solutions are available from click2learn.com. Using the eLN Connection feature in Instructor, you can utilize the ToolBook Learning Management System. For a description of this feature, see *Using the ToolBook Learning Management System*, a printed book that is part of your Instructor documentation set, and see "Building a book for use with the ToolBook Learning Management System," later in this chapter.

Ingenium from click2learn.com is a comprehensive learning management system which tracks every form of training. Progress towards achieving all types of educational goals can be measured. Online courses as well as instructor-led classes and on-the-job training can all be incorporated into a personalized curriculum. For a more detailed description of the features available in Ingenium, visit www.click2learn.com and navigate to the Products section. For information about using Ingenium with ToolBook courses, see "Managing ToolBook II courses with Ingenium," later in this chapter.

Building a book for use with the ToolBook Learning Management System

Creating a ToolBook II course that you intend to use with the ToolBook Learning Management System is essentially the same as authoring for Web delivery, with one important difference: The ToolBook II course must be self-contained, since navigation between books is not supported.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

When your ToolBook II course is completely finished and ready for distribution, you prepare the course for delivery by selecting menu options in Instructor.

➤ **To prepare your ToolBook II application for use with the ToolBook Learning Management System:**

1 From the File menu, point to eLN Connection, and then choose Setup.

2 Your default Web browser opens. Follow the instructions displayed in your Web browser.

After you receive the necessary setup information from click2learn.com, you can publish your ToolBook II application.

➤ **To publish your ToolBook II application:**

1 From the File menu, point to eLN Connection, and then choose Publish To.

2 Follow the instructions on the screen.

Depending on the options you choose in the Specialist, your browser may automatically open. Follow the instructions on the Web page displayed in your browser to finish the task of publishing your book.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Managing ToolBook II courses with Ingenium

With Ingenium, each student will launch your ToolBook II course through Ingenium Web Connect. If you plan to include questions and want the score for the answers recorded in the Ingenium database, include one of these Catalog buttons in your course: Done, or Mark as Complete. The Cancel button can be included in your course but no scoring information will be sent to Ingenium. You can include a Suspend Lesson button in your course to allow a student to save their score without finishing the lesson in one session.

When designing your ToolBook II course, you will want to refrain from creating a course structure that consists of multiple books which interact with one another (one ToolBook book calls or opens a different book). Recording the scores from this type of nested course structure is not supported in Ingenium.

After you finish building your course in ToolBook II, prepare your course for distribution by choosing Publish to Web from the File menu. This opens the Web Specialist, where you will be prompted with the following question: “Will you be using a learning management system to administer this course?” In response to this question, select the last option, which is labeled “This course will be used with an AICC-compliant server.”

When you are ready to place the exported files on a Web server for use with Ingenium, put each individual course in a different folder. For more information about using Ingenium, consult the printed and online documentation included with the Ingenium software.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Chapter 25

Printing from Instructor

Although the applications you create with ToolBook II Instructor are designed for use online, printing your application can be useful for organizing and proofing your material. You can print to any printer or output device that has been properly installed on your computer. This chapter describes how to print the pages of your Instructor application, as well as reports and scripts.

IN THIS CHAPTER

Printing pages **672**

Printing a report **675**

Formatting report text **681**

Previewing pages and reports **683**

Adding and formatting a header or footer **687**

Setting printing conditions **690**

Printing scripts **694**



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Printing pages

You can print the pages of your book to verify their organization and accuracy. Printing a draft can also help you design navigation between pages and evaluate the general page layout. You can print all of the pages in a book, pages sharing the current background, the current page, and a range of pages in a book. To set the print format, use the Print Pages dialog box.

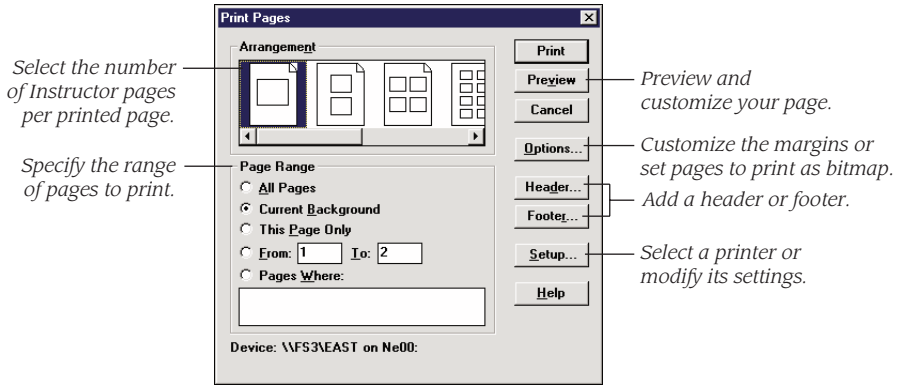


Figure 1
Print Pages dialog box

Tip To print a viewer's pages, display the book that contains the viewer's pages and then print the pages. You cannot print pages directly from a viewer other than the main window.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

➤ To print a book's pages:

Using the interface

1. From the File menu, choose Print Pages.

   The Print Pages dialog box appears.

2. To specify the number of pages you want on each printed page, choose a layout from among the graphical thumbnails under Arrangement.

   By default, Instructor prints one page per sheet of paper. If you print more than one page per sheet and the book has more than one page size, Instructor scales all pages using the largest page size. For example, if the largest page is scaled down 20 percent to fit four pages on a sheet of paper, all pages are scaled down 20 percent.

3. Under Page range, choose an option to specify which pages you want to print.

4. *Optional*. To add a header or footer to each printed page, click the Header or Footer button, type the text you want, and then click Close.

   For details, see "Adding and formatting a header or footer," later in this chapter.

5. Click Options, and then in the Print Page Options dialog box, do one or more of the following:

   - Choose a scale option. For details about page-scale options, see "Previewing pages and reports," later in this chapter.
   - If you want to print each page exactly as it appears onscreen, select Print as bitmap.
   - Select any other options you want, and then click OK to return to the Print Pages dialog box.

6. Click Print.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Using OpenScript

You can write and execute OpenScript code using the script editor or Command window. For more information about using the OpenScript programming language, refer to the *Programming in OpenScript* online book.

- To print a book's pages beginning with the current page, use the *print* command within a *start spooler* control structure:

```
start spooler
   print [<number> [pages]]
end
```

- To print all the pages in the book, use the *printall* command. The *go...print* command prints a range of pages. For example, the following script prints pages 16 through 20:

```
start spooler
   go page 16
   print 5 pages
end
```

- The *printerConditions* property prints all pages that satisfy a given condition. For example, when this property is set to the ID of the background, Instructor prints all pages sharing that background. This line should appear before *start spooler*. The background ID varies depending on how many backgrounds you have. For example:

```
printerConditions = "idNumber of this background = 1"
```

- After you have printed, you may want to clear the *printerConditions* property:

```
clear printerConditions
```




How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

- To print more than one Instructor page on a page, set the *printerArrangement* property to two positive integers, indicating the number of pages to be printed across and down a sheet:

```
--Prints 2 columns and 3 rows of ToolBook II pages
printerArrangement = 2,3
```

- To print your Instructor pages as bitmaps, set the *printerPageBitmap* property to *true*:

```
printerPageBitmap = true
```

For details about adding a header or footer using OpenScript, see "Adding and formatting a header or footer," later in this chapter. For details about scaling a page to the printer margins, setting custom margins, or printing the page at actual size, see "Previewing pages and reports," later in this chapter.



Tip If your pages have record fields, you can reorder the pages by sorting the record fields using the Sort command. For details, see Chapter 8, "Working with text."

Printing a report

With Instructor, you can print reports from the text contained in a book's record fields. For example, if you keep track of your students' names and addresses using record fields, you could print out a class address list as either a column report or a group report.

A column report prints information arranged in a table. In Figure 2, all the names are in one column, and the addresses are in the next column. Column reports are useful for inventories, check registers, parts lists, or telephone directories.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

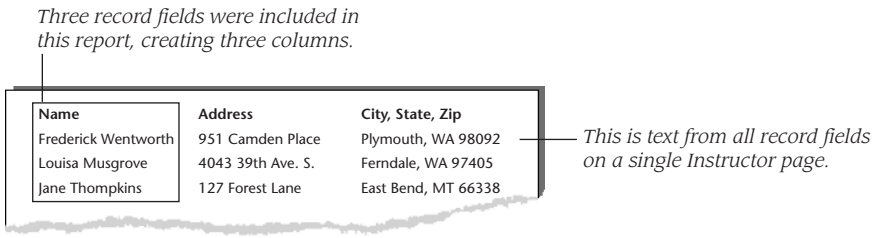


Figure 2
Column report

A group report prints all related record field information grouped together. In Figure 3, a student's name prints on one line, followed by the address on the following lines. In this case, a group is composed of all the information for a specific student that you entered on one Instructor page. Group reports are useful for mailing labels.

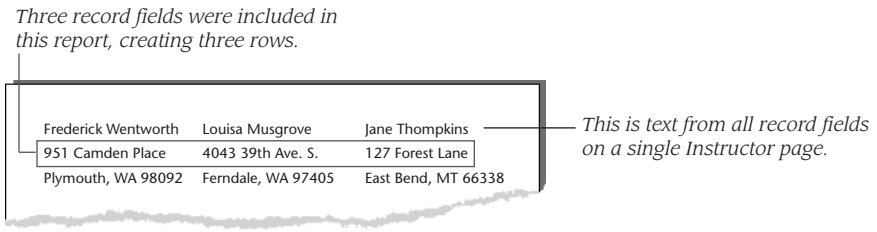


Figure 3
Group report

You can set the report format in the Print Report dialog box. Because you can print reports only from record fields, you'll need to have record fields on the current background in order to open the Print Report dialog box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

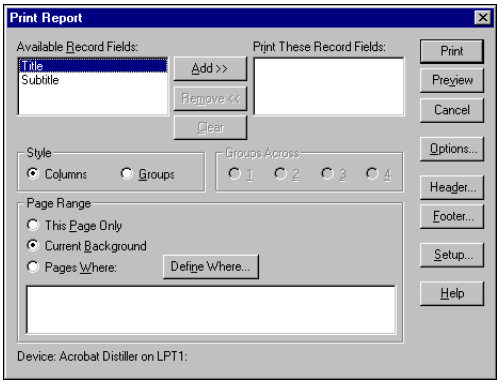


Figure 4
Print Report dialog box

For reports you print frequently, you can automate printing. For details, see “Automating printing,” later in this chapter.

➤ **To print a report:**

Using the interface

1 From the File menu, choose Print Report.

The Print Report dialog box appears.

2 In the Available record fields list, select the record field names you want, and then click Add.

3 Under Style, choose Columns or Groups. If you are printing a group report, under Groups across, choose the number of groups you want to appear across the top of the page.

How to use this online book

Contents

PART I Welcome to ToolBook II Instructor

PART II Building your application

PART III Adding interactive features to your application

PART IV Adding behavior with action sequences

PART V Using advanced features

PART VI Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

4 Under Page range, choose an option to specify the pages you want to print:

- **This page only** Prints the current page.
- **Current background** Prints all pages sharing the current background.
- **Pages where** Prints pages defined by an OpenScript statement you enter. For details, see "Setting printing conditions," later in this chapter.

5 Click Options, and then in the Print Report Options dialog box, do one or more of the following:

- To truncate text in a group report, select Clip text to fixed row height. When this option is not selected, Instructor wraps any line that is too wide to fit within the left and right boundaries of each group, creating new lines as necessary.
- Choose a scale option. For details about page-scale options, see "Previewing pages and reports," later in this chapter. Click OK to return to the Print Report dialog box.

6 *Optional.* To add a header or footer to each printed page, click the Header or Footer button, type the text you want, and then click Close.

For details, see "Adding and formatting a header or footer," later in this chapter.

7 Click Print.



How to use this online book

Contents

PART I Welcome to ToolBook II Instructor

PART II Building your application

PART III Adding interactive features to your application

PART IV Adding behavior with action sequences

PART V Using advanced features

PART VI Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Using OpenScript

- To set the report format, use these printer properties:

  printerStyle Sets the report to a *column* or *group* report.

  printerGroupsAcross For group reports, sets the number of groups printed side by side.

  printerFields Specifies which record fields are used and the order in which they appear.

  printerFieldWidths Specifies the width of columns in a column report.

  printerSize Specifies the width and height of groups. Use this property with the *printerClipText* property when you are printing mailing labels.

  printerMargins Specifies the left, right, top, and bottom margins of the printed page.

  printerGutters Specifies the vertical and horizontal distance between pages printed on a sheet, between rows in a column report, or between groups in a group report.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

- You can print a report using only a few of your record fields. For example, a database of employees might include record fields named *Name, Address, Zip, Salary,* and *Start Date.* The following handler prints a report that includes only name, address, and zip code:

```
--Report for 8 1/2-inch x 11-inch paper
to handle buttonClick
   printerStyle = groups
   printerMargins = 720,720,720,720        --1/2-inch margins
   printerGroupsAcross = 2                 --Two groups across a page
   printerGutters = 360,360                --1/4-inch width and height
   printerFields = "name,address,zip"
   printerFieldWidths = 4320               --3-inch wide groups
   --Truncate text to fit within specified margins
   printerClipText = true
   start spooler
      print
   end spooler
end buttonClick
```

- To truncate text in a group report so that it fits within the specified margins, use the *printerClipText* property:

```
printerClipText = true
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Formatting report text

Reports print using the current character formatting of record fields in your book. If you do not apply character formatting to the record fields, they print using the system font.

➤ **To format text in a record field:**

Using the interface

1 Switch to the background (press F4).

2 Select the record field you want to format. To select more than one record field, press SHIFT while clicking the record fields you want to format.

3 From the Text menu, choose Character or Paragraph to format the text.

The changes are made to all pages that share the background.



Note Text that you have formatted locally—that is, text to which you have applied formatting after selecting it in the field—will not be affected by the formatting technique described in this procedure.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Using OpenScript

- To change text character style, set the *fontFace*, *fontStyle*, or *tabSpacing* property of the record field:

```
fontFace of recordField "Title" = Arial
fontStyle of recordField "Total" = bold
tabSpacing of recordField "Sales" = .3
```

- To create an unformatted version of a record field, set the *richText* property of a record field to the *text* property of a record field:

```
richText of recordField "unformatted Subtotal" = text of \
   recordField "Subtotal"
```



Note Setting the *richText* property of a field will cause you to lose any hotwords in the text and is not reversible. To preserve hotwords, set the *fontStyle* to *regular*, *fontFace* to *system*, and *fontSize* to *10*.

If you're printing a column report, you may want to format the record field names (the names you entered in the Properties for Record Field dialog box, or the names you gave the fields using OpenScript) to use as column labels in your report.

➤ **To format report labels:**

Using the interface

1. Ensure that nothing is selected and that the cursor is not in a text field or record field.
2. From the Text menu, choose Character or Paragraph to format the report labels.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Using OpenScript

- To format report labels, set the *sysFontFace*, *sysFontStyle*, and *sysFontSize* properties:

```
sysFontFace = Arial
sysFontStyle = bold
systFontSize = 10
```

Previewing pages and reports

You can use the Print Preview window to check page layout, adjust margins, add borders, or add headers and footers.

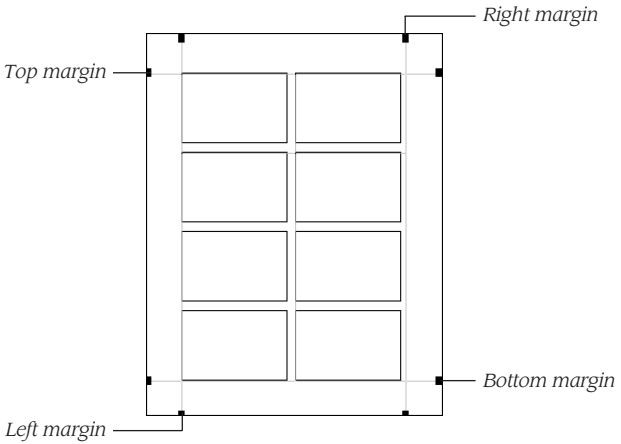


Figure 5
Margins in the Print Preview window

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

➤ **To preview a page:**

Using the interface

1 From the File menu, choose Print Pages or Print Reports, and then click Preview in the Print Pages or Print Report dialog box.

The Print Preview window appears.

2 From the Settings menu, choose Adjust Margins. If you are previewing a report, choose Draft Mode from the Settings menu to display the report's boundaries, but not its contents.

Sizing handles and guides appear.

3 To change the margins, drag the sizing handles to the positions you want. By default, Instructor scales pages to fit within the printer's margins. Alternatively, from the Settings menu, choose Options, and then choose the margins you want:

- **Actual size** Prints the pages at actual size. This option is available only when you are printing pages.
- **Scale to fit printer margins** Proportionally sizes your page to accommodate the maximum printed page allowed by your printer.
- **Scale to fit custom margins** Proportionally sizes your page to fit between the left, right, top, and bottom margins you specify.

Click OK to close the Print Page Options or Print Report Options dialog box.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

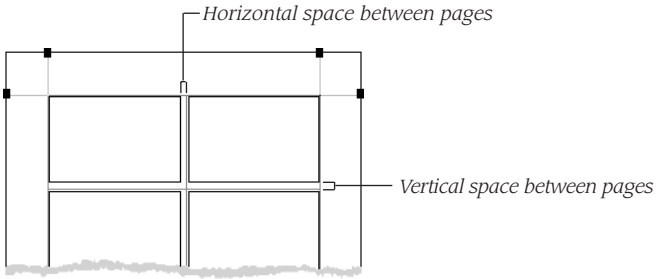
25 Printing from Instructor

Glossary

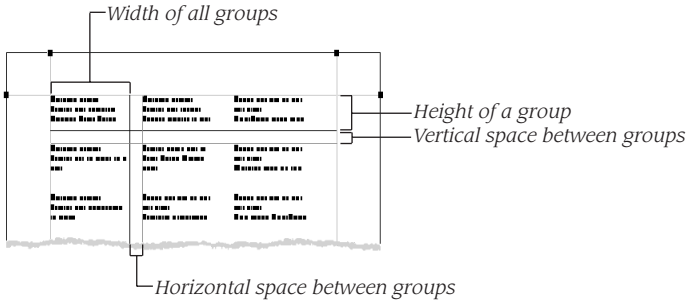
Index

4 To change the layout of your pages, drag the guides to the position you want.

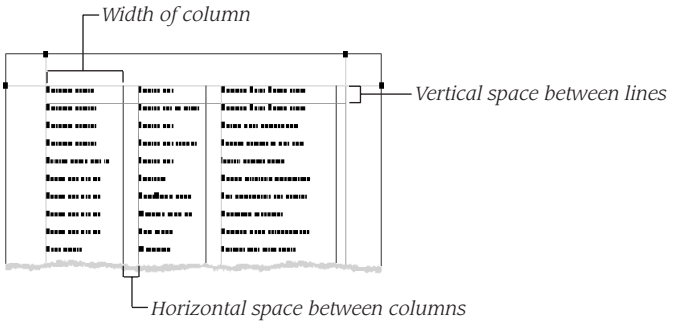
Print pages



Group report



Column report



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

5 If you are printing pages and want to print a border around each Instructor page, from the Settings menu, choose Options, select Print borders around pages, and then click OK.

6 When you are ready to print, from the File menu, choose Print.

To view the Print Pages or Print Report dialog box again, from the File menu, choose Exit Preview.



Tip To make it easier to see and select the handles and guides, from the Settings menu, choose Zoom.

Using OpenScript

- To set the page or report to print at actual size, to scale to the printer margin, or to scale to a custom margin, set the *printerScaling* property to *actual*, *printer*, or *custom*. (*Actual* is available only for printing pages.) By default, the *printerScaling* property is set to *printer*. For example:

```
printerScaling = actual
```

- To adjust the margins and guides, use the *printerMargins* property to set the left, right, top, and bottom margins in page units. Use the *printerGutters* property to set the width and height of the gutter in page units. For example:

```
printerMargins = 1440,1440,1440,1440      --Sets all margins 1 inch
printerGutters = 333,333
```

- To add a border around each Instructor page, use the *printerBorders* property:

```
printerBorders = true
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Adding and formatting a header or footer

You can add a header or footer to your printed pages to convey page numbers, the date and time the pages were printed, your application name, the Instructor file name and location, or a message such as "Confidential Draft."

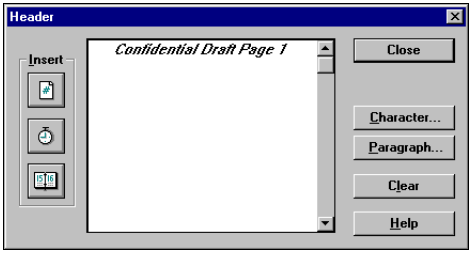


Figure 6
The Header dialog box

You can modify the character formatting (such as font and size) and paragraph formatting (such as spacing, alignment, and indents) for headers and footers.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

➤ To add and format a header or footer:

Using the interface

1. Choose Print Pages or Print Report from the File menu, and then click the Header or Footer button.

   The Header or Footer dialog box appears.

2. Add the information you want in the header or footer:

   - Type the header or footer text in the Header or Footer box. Press CTRL+TAB to insert tabs; press CTRL+ENTER to start a new line.
   - Insert page numbers, the time, or the date by clicking one of the options under Insert. Each time you print, the time and date are updated to reflect the current time and date.

3. To format the header or footer text, click Character or Paragraph, choose the options you want for the text, and then click OK.

4. Click Close to close the Header or Footer dialog box.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Using OpenScript

- To add a header or footer to a printed page, use the *header* or *footer* book property, which specifies the contents of the header or footer:

```
header of this book = "April Status Report"
footer of this book = "April 15, 2000"
```

- You can set the *header* or *footer* property to a string of up to 32,000 characters. To specify a print time, print date, or running page number, enter *~t* (time), *~d* (date), or *~p* (page number):

```
header of this book = "Personnel Records as of ~d page: ~p"
```

- You cannot locally format header or footer text using OpenScript. To format the header or footer text, use a script like the one below:

```
--Saves the old font settings
saveFace = sysFontFace
saveStyle = sysFontStyle
saveSize = sysFontSize
clear header of this book

--Sets the desired font, size, and style
sysFontFace = "Times New Roman"
sysFontStyle = "bold,italic"
sysFontSize = "10"
header of this book = "Module 1"

--Restores the font settings
sysFontFace = saveFace
sysFontStyle = saveStyle
sysFontSize = saveSize

--Prints the header and settings you defined
start spooler
   print all
end
```

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Setting printing conditions

Using OpenScript expressions, you can define conditions that control which pages in a book are printed. For example, if you have a book that contains repeating chapter sections, you can print just the pages for one chapter. When defining printing conditions, the expression can be a valid statement that evaluates to *true* or *false*. If the expression is *true* for a page, the conditions are met and the page is printed. If the expression is *false* for a page, it is not printed.

➤ **To set printing conditions for pages:**

1 From the File menu, choose Print Pages.

The Print Pages dialog box appears.

2 Under Page range, choose Pages where, and then enter an OpenScript expression.

For example, to print all pages where the record field named *Chapter* includes the text *geothermal energy*, type this expression:

```
text of recordField "Chapter" contains "geothermal energy"
```

Or, to print all pages where the record field named *Total* includes a number over 100, type this expression:

```
text of recordfield "Total" > 100
```

If the conditions you specify are not satisfied by any of the pages in the book, Instructor displays a message when you click Print or Preview.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

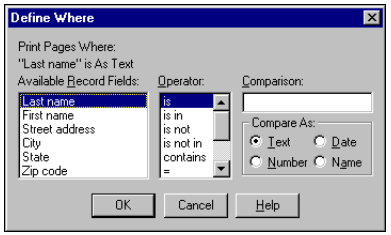
➤ To set printing conditions for reports:

1 From the File menu, choose Print Report.

The Print Report dialog box appears.

2 Click Define Where in the Print Report dialog box.

The Define Where dialog box appears.



3 In the Available record fields list, select the record field name you want to compare.

4 In the Operator list, select the comparison operator for the condition you want to define.

5 In the Comparison box, type the value for completing the criteria for printing. This value will be compared against the contents of each record field; only those that match the value will be printed.

If necessary, choose the type of data that you are comparing under Compare as.

If the conditions you specify are not satisfied by any of the pages in the book, Instructor displays a message when you click Print or Preview.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Example printing conditions

The following table lists examples of conditions defined in OpenScript.

Example printing conditions

To do this...	Use this condition...
Print pages for which the content of the *Material* field is the text *plastic*	`text of field "Material" is "plastic"`
Print pages for which the content of the *Summary* field is the text *solar battery* or *solar cell*	`text of field "Summary" contains "solar"`
Print pages for which the content of the *Material* field is not *Styrofoam*	`text of field "Material" is not "Styrofoam"`
Print pages that are named *Introduction*	`name of this page is "Introduction"`
Print pages that use a background named *Presentation*	`name of this background is "Presentation"`
Print pages that contain at least one field	`objects of this page contains "field"`

(continued)

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

Example printing conditions, *continued*

To do this...	Use this condition...
Print pages for which *Delegate* record field contains *Musgrove,* and *Contribution* record field contains number greater than or equal to 1000	`text of recordField "Delegate" contains "Musgrove" \ and text of recordField "Contribution" >= 1000`
Print pages for which last name in *Student* record field starts with a letter between N and Z, inclusive	`text of recordField "Student" > "M" as name`
Print pages for which *PurchaseDate* record field contains today's date	`text of recordField "PurchaseDate" = sysDate as date`



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

23 Preparing applications for release

24 Distributing applications on the Internet

25 Printing from Instructor

Glossary

Index

In the following handler, Instructor prints a group report listing managers hired in 2000:

```
to handle buttonClick
   printerFields = "First name,Last name,Title,Hired"
   printerStyle = groups
   --Limits the report data to managers hired in 2000
   printerConditions = "text of recordField ""Title""" && \
      "contains ""manager""" and text of recordField ""Hire Date""" && \
      "contains ""2000"""
   …     --Further statements here
   start spooler
      print all pages  --Prints only pages that meet the conditions
   end spooler
end buttonClick
```

Printing scripts

Printing scripts is a good way to keep track of them and to check for errors. Scripts are printed without headers, footers, or page numbers.

Instructor truncates lines longer than 80 characters. To keep long statements from running off the page when they are printed, add a backslash (\) to continue statements on additional lines.

➤ **To print a script:**

1 Display an object's script in the script editor.

2 In the script editor, choose Print from the File menu.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Glossary

&, && *See* concatenation operators.

access character An underscored character on a button, menu, or menu command that provides direct keyboard access to the option. Press the ALT key in combination with the access character to select the button, menu, or menu command. *See also* button, label button.

action A single, complete instruction that you create in the Actions Editor to add behavior to your application. In the Actions Editor, you can specify actions, conditions, and loops, and this behavior will automatically export to DHTML when you use the ToolBook II Web Specialist. *See also* Actions Editor, action sequence, condition, DHTML, loop.

action sequence One or more behaviors that you create in the Actions Editor that Instructor carries out in the order you specify. Unlike an OpenScript program, which requires the Neuron plug-in to function on the Internet, an action sequence will export to DHTML, an Internet technology that runs in a Web browser. *See also* Actions Editor, DHTML, ToolBook II Web Specialist.

Actions Editor Instructor's visual programming tool. In the Actions Editor, you use familiar interface elements such as menus to add behavior to your Instructor application, and this behavior will export to DHTML. *See also* action, action sequence, DHTML, shared action sequence, ToolBook II Web Specialist.

activated A property of a field or record field that enables the field to perform an action when a user clicks it. When a field is activated, a user cannot enter text in it. *See also* field, record field.

ActiveX control A software component that you can include in your Instructor application to add custom functionality. You can respond to an ActiveX event or manipulate its properties and methods in an action sequence or an OpenScript program.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

ActiveX Data Objects (ADO) A high-level mechanism that allows you to access the data in databases from Instructor using OpenScript. *See also* Automation.

AICC *See* Aviation Industry Computer-Based Training Committee.

alias (1) An alternative name for a menu item that is not visible to the user. If you create an alias for a menu item, Instructor sends the alias in place of the menu item name. (2) When linking a dynamic-link library function, an alternative name given to the function. Use an alias to avoid conflicting with Instructor keywords or when you want to link the same function more than once in the same instance of Instructor. *See also* dynamic-link libraries.

American National Standards Institute (ANSI) Typically, ANSI refers to a set of 256 characters used in computing. The first 128 of these characters are the same as the ASCII character set. The remaining characters (128-255) include non-English letters and other symbols called extended characters. *See also* American Standard Code for Information Interchange.

American Standard Code for Information Interchange (ASCII) Typically, ASCII refers to the 8-bit (256-character) standard for representing characters (both printing and nonprinting) on a computer. For example, the ASCII representation (in decimal) for a space is 32. *See also* American National Standards Institute.

analog Information that flows in a continuous stream and that can vary in infinitesimal increments. Most computers cannot manipulate analog data, so it must be converted to digital format by breaking the continuous flow into discrete steps and assigning numeric values to each step.

angled line A single line with three or more points, such as a zigzag. The Catalog contains angled lines that you can drag and drop onto a page.

animation A file that you can incorporate as a clip in Instructor. You can also animate objects in a book; however, you cannot save an animation created in Instructor as an animation file or create a clip from it.

ANSI *See* American National Standards Institute.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

application In Instructor, one or more books designed for a particular purpose, such as online learning or information management.

argument A constant, variable, or expression passed to a function or handler. In the following example, 33 is an argument:

x = sin(33)

See also constant, function, handler.

array A variable configured to work like a list or table of individual elements. For example, the following specifies the value of one element in an array:

x[3] = x[1] & text of field "names"

See also variable.

ASCII *See* American Standard Code for Information Interchange.

audio CD Sound stored as tracks on a compact disc and played from the compact disc. Audio CD is stored in digital format (as a series of binary values), converted to analog sound within the CD-ROM drive, and then played back.

Audio Video Interleaved (AVI) A format for storing digital video in a file that alternates blocks of visual and sound information. AVI files are particularly suited for storing on CD-ROM because their interleaved format allows them to be efficiently retrieved by CD-ROM drives.

Author level One of two working levels in Instructor. Author level provides tools and commands for creating and modifying objects, including pages, books, and backgrounds. Applications are built at Author level and run at Reader level.

Automation A technology that allows you to access and manipulate the objects, properties, and methods of other Windows applications from an Instructor application.

AVI *See* Audio Video Interleaved.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Aviation Industry Computer-Based Training committee (AICC) An industry standards group that publishes guidelines for interaction between course management systems and computer-based training.

backdrop A bitmap image placed onto one or all of a book's backgrounds. Each backdrop becomes a property of its background.

background A design shared by pages in a book. Every book has at least one background. Objects on a background appear on every page sharing that background. Several pages can share the same background, and a book can contain several backgrounds.

bandwidth The amount of data that can be sent via the Internet over a period of time with the equipment being used. Bandwidth can refer to the connections on the Internet, as well as the connection between a modem and an Internet service provider.

baselines Lines that show the vertical spacing of text in a field or record field. *See also* field, record field.

bitmap Also known as a paint object. A graphic created in a painting program and stored as a map of the pixels that make up the image. Windows bitmaps generally have .bmp or .dib file extensions. Bitmaps can be pasted or imported into Instructor or assigned as book resources. *See also* resources. Compare draw object, picture object.

book A collection of Instructor pages stored together as a single file. A book can be an entire application or one part of an application.

book palette In 256-color systems, a predefined list of colors assigned to a book. If a book has a palette, all imported graphics displayed in the book use the book palette, which overrides the palettes of the individual paint objects. Assigning a book palette eliminates palette flash, which occurs when you navigate to a page containing an imported graphic that has a palette different from the one currently in use.

book properties Attributes of a book that define its appearance and behavior.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Book Specialists The wizards that Instructor provides to guide you through the process of creating a new book. Instructor includes three Book Specialists: the Full Specialist, the Lesson Design Specialist, and the Quick Specialist.

Boolean Refers to properties and variables whose values can only be true or false. Referred to as logical in OpenScript.

border (1) The boundary of a field, record field, or button, or the line enclosing a graphic. (2) A line that Instructor adds around each page when the page is printed. *See also* border style.

border style A property of a button, graphic, field, or record field that determines the style of the object's border.

bounding box The rectangle around a paint or picture object that indicates its dimensions and the area in which it will display. The bounding box has a thin border, but you can make the border invisible.

breakpoint A flag set in a statement while debugging a script that indicates where Instructor should invoke the Debugger when executing that statement. Breakpoints are set and cleared in the Debugger. See also Debugger, statement.

browser A software program that accesses Web pages on the Internet. The browser reads the Web document and translates it into the final product onscreen. The page appears as the author intended, combining text, graphics, and jumps to other Web pages.

built-in message A message built into the Instructor system that Instructor sends in response to user activity (for example, buttonClick). Compare user-defined message.

button An object type that includes pushbuttons, command buttons, check boxes, radio buttons, and label buttons. Buttons are often used to trigger events or set properties.

call A reference in a script to another handler. *See also* handler, script.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

called handler The handler that executes as the result of sending a message, invoking a user-defined function in an expression, or using the set or get command. *See also* handler, user-defined function.

calling handler Any handler that makes a call, causing the execution of another handler. In OpenScript, send and forward statements call other handlers. *See also* handler.

caption The text that appears in the title bar of a window, on a button, or in the title bar of a book in place of the file name.

caret location The location of the insertion point within a text field or record field, measured by the number of lines down from the top and number of characters in from the left. The caret location is often used to define the ends of lines in a text field. A line is designated by a carriage return/linefeed, not by wordwrap.

carriage return/linefeed (CRLF) An OpenScript constant that represents ASCII values 13 and 10, the characters carriage return and linefeed. CRLF moves text to the next line in a text field or record field.

cascading menu A subsidiary menu that appears to the side of a submenu and presents additional choices.

Catalog An Instructor file that contains authoring objects, such as draw objects, action buttons, question objects, graphics, media players, navigation objects, and user-customized objects.

cel animation A type of animation in which individual views of an object, called cels, are shown in rapid sequence to make the object appear to move or change. In Instructor, you can create cel animations by drawing individual objects to act as cels and then grouping them.

check box A button style that consists of a square box and a button caption. A check mark toggles on and off when the check box is clicked at Reader level; when checked, the option is turned on.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

check mark The symbol that appears next to a menu item or in a check box indicating that the option is selected. Choosing the menu item again or clicking a checked check box turns the option off and hides the check mark. *See also* check box.

child object In Instructor, an object that is owned by another object. For example, a button on a page is a child object of the page.

child window A window that is entirely contained within and cannot be dragged beyond the boundaries of its parent window. A child window can be minimized or tiled within its parent and closes when its parent window closes. *See also* pop-up window.

chromakey A feature that allows you to create transparent areas in bitmaps by specifying a color to make invisible.

client application An application that can accept linked and embedded objects from a server application. Instructor is a client application. See also server application. Compare object linking and embedding.

client area The area inside the frame window, including the client window. *See also* client window, frame window.

client window The portion of the client area that displays pages and backgrounds. *See also* client area.

client-side image map An image map for which the map information is specified in the HTML file that contains the image. Clicks are processed by the Web browser without interaction with a server.

clip A reference to a segment of any media source or file, including sound files, audio CD, digital video files, animation, or videotape. Only the reference is stored in Instructor. Instructor supports file-based clips, including sound files, digital video files, and animation files.

clip library A file (extension .cpf) that contains a collection of clips (references to media files) that can be used in one or more applications and that can refer to different types of media. *See also* clip, media file.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

clip text A printing option to cut off text at a field's margin or at the boundary of a column or row when printing a report.

Clipboard A temporary storage location where Windows places the item most recently cut or copied. The Clipboard can be used to move text and graphics between Instructor books and other Windows applications.

Coach A quick reference tool that provides context-sensitive information about pages and objects. You can open the Coach from the Instructor Help menu.

color A combination of hue, lightness, and saturation (HLS) or red, green, and blue values (RGB). Objects created in Instructor can have a stroke color and a fill color, which are chosen from a color palette or the Color dialog box. *See also* hue, lightness, and saturation; red, green, and blue values.

color palette A predefined set of colors stored in a palette file (with the file extension .pal). Instructor uses a 256-color palette. Visual media—animations, videos, or bitmaps—have their own color palettes.

color tray An Instructor system object used to assign stroke and fill colors to objects.

column report A report printed in columns; each column contains the contents of one record field. *See also* record field. Compare group report.

combo box A single object composed of an edit box, a pushbutton, and a drop-down list used to present a list of choices in an application. A combo box shows only the selected item. To display all items, the user clicks the pushbutton. *Compare* list box.

command (1) A menu item that, when chosen, causes Instructor to perform an action. (2) A term in the OpenScript language. Most OpenScript commands are verbs, such as go, move, and ask. *See also* Command window.

Command window The window in which you can run script statements, get information about objects and system variables, and set object properties. *See also* script, system variable.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Command window history The Command window contains a split bar that can be dragged down to display the last 20 commands entered in the current session. Click a command to reinsert it in the Command window.

comment Optional information the programmer adds to a script that does not cause Instructor to take any action. Comments start with two hyphens. For example:

```
--Sets the object's fontFace
fontFace of field "This Field" = "Times New Roman"
```

compiled path A property of an animated object containing the coordinates for the object at each step of the animation. Instructor plays the animation by using the next set of coordinates (four coordinates per step) from the compiled path and setting the object's bounds to them. The number of steps in the compiled path is based on the duration and steps per second specified when you saved the animation in the ToolBook II Animation Editor. A compiled path with 30 steps (including the start and end points) will have 120 items in its compiled path.

concatenation operators The symbols in OpenScript (& and &&) that instruct the application to join two strings of characters into a single, longer string. The double ampersand operator (&&) adds a space between the joined strings. For example:

```
vMessage = "Hello," && "world"
```

condition A special type of action that, when inserted into an action sequence, creates a junction at which Instructor follows one of two or more distinct paths as it executes the action sequence. *See also* action, action sequence, loop.

constant A value that cannot change, including literal numbers and strings. For example, in the expression x + 2, 2 is a constant. See also literal value, string. Compare variable and container.

constrain The action of limiting the proportions, angles, or movement of an object when you create or move it on a page or background.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

container Anything that can be set to a value using OpenScript. Variables and properties are containers. Constants (such as the string constant "blue") and expressions (such as 1 + 2) are not containers because their values cannot be set. *See also* constant, properties, variable.

control character A nonprinting character, such as a tab or a carriage return/linefeed, that affects the appearance of text. *See also* carriage return/linefeed, tab.

control structure A block of statements in a script that is run under particular circumstances, such as in a loop or as a condition. Control structures are defined by sets of OpenScript keywords such as if/then/else/end, do/until, and condition/when/end. *See also* script, statement.

controller card A board installed in a computer that provides a hardware link between the central processing unit and devices such as disks, monitors, or speakers.

CRLF *See* carriage return/line feed.

crop To mask or hide part of a paint object. When you drag the handle of a paint object to make it smaller, the paint object does not resize; instead the area is cropped. When you resize a picture object or draw object, the entire object is resized and no part of it is cropped. *See also* draw object, paint object, picture object.

cursor The image assigned to the pointer (select tool, insertion point, I-beam, crosshair), depending on where the pointer is positioned and whether a tool is selected. *See also* pointer.

cursor resource A cursor image that can be imported and shared as a resource in one or more Instructor applications. *See also* cursor, resources.

custom colors In 256-color systems, a list of 64 colors that are available beyond the standard 16 or 32 colors.

data field An individual entry in a data record in an imported or exported fixed-field file or delimited-field file. Often a delimiter character, such as a tab or comma, separates data fields from each other in data records. See also delimited-field file, delimiter, fixed-field file.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

data typing Assigning a format to a variable in a script that determines what type of data the variable will accept (integers, floating-point numbers, character strings, logical values, and so on). *See also* variable.

Debugger The window that displays script errors at Author level. The Debugger can be used to step through a script one statement at a time to isolate and correct errors. *See also* script, statement.

default A setting that Instructor supplies (for example, the character default is System, bold, 10 point). Instructor has a default setting for any changeable setting. All defaults are system properties. *See also* system properties.

delimited-field file A text file divided into data fields of any length that are separated by a delimiter character such as a comma or a tab. All the data fields in one line make up a data record. *Compare* fixed-field file.

delimiter A character such as a comma or tab that separates one data field from another in a delimited-field file. *See also* delimited-field file.

destination object In a drag-and-drop operation, the object that accepts a drop from an object that is dragged (the source object). *See also* drag and drop.

destination page The target page of a hyperlink. *See also* hyperlink.

device channel The hardware path by which data is passed to the output device that plays it. For example, a sound card contains a device channel for sending audio information to speakers.

device driver A program that allows Windows to communicate with a specific piece of multimedia hardware. Device drivers are installed through the Windows Control Panel. *See also* Media Control Interface.

device-independent bitmap A bitmap that appears exactly the same way regardless of the device used to display it. These files have a .dib file extension. *See also* bitmap.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

DHTML *See* Dynamic Hypertext Markup Language.

digital A means of describing information (pictures, sound, video, or any other type of information) using a series of numbers. For example, a photograph can be described digitally by treating it as a series of very small dots and recording the position, color, and lightness of each dot as a number. Using the appropriate process, the numbers can be passed to a program that can use them to recreate the dots and thereby recreate the photograph.

digital video Video converted into a digital format and stored as a file.

display device The combination of video card and video driver that determines a system's color capability.

dithered color A color that appears on the screen as a mixture of different solid colors that attempts to approximate the original color.

DLL *See* dynamic-link library.

drag and drop The process of clicking an object, dragging to another location on the page, and releasing the mouse. Drag and drop is often used as an intuitive visual interface for such operations as moving or deleting objects.

drag image The image that is displayed during a drag-and-drop operation when the pointer is over a potential destination object. *See also* destination object, drag and drop.

draw direct Drawing an object directly on the screen rather than on an offscreen bitmap that is then transferred to the screen. By default, new objects are drawn directly. *Compare* offscreen image.

draw object In Instructor, draw objects are the arc, angled line, curve, ellipse, regular and irregular polygon, line, pie, rectangle, and rounded rectangle. *Compare* paint object, picture object.

draw-type graphic Also known as a vector graphic. An illustration created in a drawing program that maps the shapes you draw on an invisible grid and stores them as a set of instructions. *Compare* bitmap.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

drop-down menu A menu that descends from the menu bar when clicked. For example, when a user clicks File in the menu bar, the File drop-down menu descends.

Dynamic Data Exchange (DDE) A communication protocol to exchange data and commands between Windows applications.

Dynamic Hypertext Markup Language (DHTML) A technology that uses tags to specify the properties and behavior of an HTML document. DHTML extends functionality available in HTML and enables interactive behavior for such features as buttons, animations, actions, and responses. Instructor can export an application to DHTML by using the ToolBook II Web Specialist. *See also* ToolBook II Web Specialist.

dynamic-link library (DLL) A separate function library that a Windows application can call to perform tasks. DLLs are used to add capabilities not available directly from Instructor.

editable A property of combo boxes, text fields, and record fields that specifies whether a user can type text at Reader level. *See also* text field, record field.

embedded object An object that has been embedded in another application (such as Instructor) using object linking and embedding (OLE) technology.

enabled A property of buttons, text fields, and record fields that determines whether an object can receive the focus or mouse-event messages. When an object is enabled, it is included in the tab order and can receive the focus or mouse-event messages as a result of keystrokes and mouse clicks. *See also* button, text field, record field, tab order.

enclosed object An object that is closed on all sides and that can be filled with a color or pattern. The following objects are enclosed: rectangle, rounded rectangle, ellipse, polygon, irregular polygon, and pie.

end of file (EOF) An OpenScript constant that represents ASCII value 26, a character often used to mark the end of a DOS file.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

error box A window that appears in the Debugger and describes why a script's execution was suspended. *See also* Debugger, script.

event An action recognized by Instructor, such as the clicking of a button or the pressing of a key. Events can be initiated by user action, by sending the generic User event in an action sequence, or by an OpenScript command.

event focus properties Instructor system properties that indicate what object is currently receiving messages, such as target, focus, targetWindow, focusWindow, and selection.

execute To carry out actions defined by OpenScript statements or actions created in the Actions Editor. *See also* statements.

Execution Suspended message A message that contains details about why and where an error occurred in a handler. The object that was handling the message when the error occurred is also specified. *See also* handler.

explicit reference A reference to an object that explicitly identifies its parent object. Explicit references are used to refer to objects that are not on the current page. For example, the following statement explicitly references the button Next:

```
get caption of button "Next" of background "lessons" of book "Tutorial"
```

See also parent object. Compare implicit reference.

export To transfer data from one file format to another. For example, you can export your book to HTML, or you can export text to ASCII format. *See also* import.

expression A combination of values that yields a result, such as the name of a field, variable, property, or an expression with operators and operands. For example, 7 + 3 is an expression yielding the result 10. See also field, operand, operator, properties. *Compare* container.

extended character A non-English character or symbol in the ANSI character set. Extended characters have ANSI values of 128 to 255.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

extended properties Properties that specify the extended capabilities of such objects as question objects and action triggers. *Compare* properties.

external media Any media file or device referred to but not stored within an application, such as a videodisc.

feedback A feature of online learning used to provide responses or replies to users according to how they answer a question.

field An object that holds text. Instructor uses text fields and record fields.

file path The string of characters and delimiters that specify the location of a file, as in the following example:

C:\Program Files\ToolBook\Instructor\Inst.exe.

File Transfer Protocol (FTP) A set of rules for transferring files from one computer to another. Instructor includes an FTP utility you can use to transfer files from one computer to another.

fill The color or pattern applied to the inside of any enclosed object. You cannot apply a fill color to paint objects or picture objects. *See also* paint object, picture object, stroke.

filter A utility that allows Instructor to import and convert files from other applications.

fixed-field file A text file in which each data field consists of a predefined number of characters. *Compare* delimited-field file.

focus The object awaiting the next keyboard action. Pressing the TAB key moves the focus. Only objects that can respond to keystrokes can receive the focus. The insertion point in a field or a dotted rectangle in a button shows a user the current focus.

focus window The viewer that currently has the focus. The focus window is the viewer that accepts mouse and keyboard input (such as selecting check boxes or filling in text fields). *See also* focus, viewer.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

font In Windows, a font is the complete collection of styles and sizes for a single typeface (such as Times New Roman) as part of a set installed for a system's display adapter, graphics card, and printer.

footer Text that appears at the bottom of every printed page.

frame window The portion of a window that contains the border, caption bar, system menu, scroll bars, and Minimize and Maximize buttons. The frame window surrounds the client area. *See also* client area.

FTP *See* File Transfer Protocol.

Full Specialist A wizard that gathers input from you and then builds a book based on your specifications. The Full Specialist, the Quick Specialist, and the Lesson Design Specialist are Book Specialists that guide you through the process of building a new Instructor book. *See also* Lesson Design Specialist, Quick Specialist.

function An operation that uses a predefined formula and returns a value. For example, the function *round* always returns the supplied value, which may be a literal value or expression, rounded to the nearest integer.

function keys The keys on a computer keyboard labeled F1, F2, F3, and so on. Function keys can be used independently or together with the CTRL, ALT, or SHIFT key as shortcuts for choosing menu commands and performing other tasks.

general system properties Properties of Instructor that affect every book opened during the current Instructor instance. For example, Instructor uses the value of the property sysDateFormat to format all dates.

global memory The area of memory available to all Windows applications.

global variable In an action sequence, a variable whose value can be accessed and used in any action sequence. *See also* variable, system variable. *Compare* local variable.

graphic A paint object or picture object placed on an Instructor page. *See also* paint object, picture object.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

grid A set of nonprinting dots used at Author level to precisely align objects on a page.

group A collection of objects that is defined to move and be resized as a single object. Like any other Instructor object, a group has an ID number and has properties that can be set.

group report A report printed in groups, where each group contains the contents of the record fields from one page. For example, a group report can print mailing labels from an address book. See also record field. *Compare* column report.

gutter On a printed report, the space between adjacent columns in a column report, groups in a group report, or Instructor pages (if you are printing more than one page per sheet of paper). *See also* column report, group report.

handle (1) A small square that appears at each corner and edge of an object when it is selected. Handles are visible only at Author level. You can drag the handles to resize an object. (2) A unique reference number assigned by Windows to any open viewer, whether it is shown or hidden. A handle can also refer to a memory location. *See also* viewer.

handler A collection of OpenScript statements in a script that defines the response to a particular event, such as clicking an object. A script can contain more than one handler, each responding to a different event. *See also* script, statement.

header Text that appears at the top of every printed page.

highlight (1) The onscreen indication that text or another item is selected. (2) The property of a button or hotword that controls whether it flashes when a user clicks it. *See also* button, hotword.

home page Typically refers to either the first Web page in a series or to the Web page specified to open automatically with your browser. *See also* Web page.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

hotword An object made up of one or more words in a text field or record field that has an ID number and can respond to mouse and keyboard events.

HTML *See* Hypertext Markup Language.

HTTP *See* Hypertext Transfer Protocol.

hue, lightness, and saturation (HLS) Hue is the quality of a color corresponding to its position in the spectrum: red, orange, yellow, green, cyan, blue, magenta. Lightness is the amount of white or black in a color. Saturation is the intensity of a color. At 0 (zero) percent saturation, a color appears white; at 100 percent saturation, a color's hue is most intense. *See also* red, green, and blue values (RGB).

hyperlink A property that links objects to another page or URL. A hyperlink can be assigned to hotwords, buttons, or any other Instructor object. *See also* button, hotword.

hypertext A word or group of words that, when clicked, cause the browser to perform an action, such as navigation to another page. *See also* hyperlink, hotword.

Hypertext Markup Language (HTML) A standard document format used for pages on the World Wide Web that uses tags to specify the properties of the document. A Web browser can interpret the HTML tags and show the formatted document on the screen. Instructor can export an application as a series of HTML pages.

Hypertext Transfer Protocol (HTTP) A standardized procedure for transferring information via the World Wide Web. One of several information transfer protocols used on the Internet.

I-beam An I-shaped cursor that appears over locations where text can be edited. *See also* cursor.

icon A graphic that represents an application.

icon resource An icon that can be imported and shared as a resource in one or more Instructor applications. *See also* resources.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

ID number A unique identification number that Instructor assigns each object when it is created. Each copy of an object has an ID number that is unique for the page on which it is located. When an object is cut and pasted it is assigned a new ID number. *See also* unique identifier.

image buffer An area of memory in which Instructor maintains an image of the current page or background of a viewer in order to display it quickly when required.

implicit reference A reference to an object that identifies it without stating its explicit location; the object is assumed to be on the current page. For example, the following statement refers to a button on the current page:

```
get caption of button "Next"
```

Compare explicit reference.

import The process of bringing data such as text, graphics, and objects created in another application into Instructor. *See also* export.

indent The space between the left and right boundaries of a field and the left and right edges of text in a field.

initialize To assign a starting value to a variable. *See also* variable.

insertion point A blinking cursor that shows where the next text entered will appear. *See also* cursor.

instance A open copy of any Windows application, appearing as either a window or an icon. You can run multiple instances of Instructor in order to work with more than one book or page at the same time.

international system properties Instructor properties containing format information for data that changes according to country and language, such as sysCurrency and sysDecimal, which specify what characters are used for these functions.

Internet A network of networks where information is shared and exchanged using agreed-upon hardware and software communications protocols.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

irregular polygon An enclosed draw object with unequal angles and sides of varying length. *Compare* polygon.

item One data element out of a string separated by commas. The item does not include the comma, which is used only as a delimiter. *Compare* textline.

label button A button style composed of text with an access character. The user presses the ALT key in combination with the underlined letter in the button's caption to achieve a button click. Often used for labels containing read-only information and for dialog box controls. *See also* button, focus, access character, tab order.

layer The relative order of all the objects on a page or background. The most recently created object is on the front layer. You can change an object's layer to control the way objects overlap one another and to define the tab order. *See also* tab order.

Lesson Design Specialist A wizard that helps you build a lesson specially tailored to meet your teaching goals. The Full Specialist, the Quick Specialist, and the Lesson Design Specialist are Book Specialists that guide you through the process of building a new Instructor book. *See also* Full Specialist, Quick Specialist.

line spacing The space between individual lines of text in a field. Line spacing can be set to single, double, or one-and-a-half lines. A field's line spacing is visible only when its baselines are displayed. *See also* baselines, text field.

line style The visual appearance of a line (solid, dashed, dotted), determined by choosing a style in the Properties dialog box.

list A group of one or more elements separated by commas. Instructor uses lists for many data structures (such as the coordinates of a cursor or a bounding box) and has special OpenScript commands to manipulate them (pop, push, and so on).

list box A type of field used to display a list of choices; list boxes can be defined as single-select or multi-select. Unlike combo boxes, a list box can display more than one option at a time. *Compare* combo box.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

literal value A string of characters or a number that is assigned to a container or used to compare information. If a literal value contains spaces, it must be enclosed in quotation marks. In the following example, New York is the literal value:

```
text of field ID 1 = "New York"
```

See also container.

local variable In an action sequence or script, a variable whose value can only be accessed and changed as the script or action sequence in which it is defined is running and which only maintains its value while the script or action sequence is running. *See also* variable. *Compare* global variable, system variable.

log file A file maintained by Instructor that stores information about a user's progress in a lesson, such as the pages visited, the user's score, the time, the date, and the elapsed time. You can use the log reader utility to view a log file.

logging The process of recording user activity in a lesson. Logging starts when the user enters the lesson and stops when the user leaves the lesson.

logical palette In a 256-color system, the list of colors required to display a graphic. Each graphic maintains its own logical palette of 256 colors from the 16 million colors possible.

loop A special type of action that, when inserted into an action sequence, directs Instructor to repeat a behavior (an action or sequence of actions) a specified number of times or until a condition is met. *See also* action sequence, action, condition.

margin (1) The space between a field's boundaries and the text in the field. (2) The space between the edge of the paper and printed text or graphics. *See also* indent.

mat A color border that appears if a page is smaller than the window in which it is displayed.

MCI *See* Media Control Interface.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Media Control Interface (MCI) A Windows-standard set of commands used to control media devices. *See also* media device.

media device Hardware that controls a source of multimedia (for example, a CD-ROM drive that can play an audio compact disc). *See also* audio CD, multimedia.

media file A file that contains media, such as a digital video file or a bitmap.

media player A Catalog object that is preprogrammed to play media. The Universal Media Player is a multi-purpose media player that supports most currently available media formats.

media source Any multimedia from which you can create clips, including media files, audio CD, videodiscs, or videotape.

menu A list of choices that a user can click to open windows, execute commands, or display further choices.

menu bar The area at the top of a window that lists available menus.

Menu Bar Editor The Instructor editor used to create and edit menu bars.

menu bar resource A menu that can be shared as a resource in one or more Instructor applications. *See also* resources.

menu item A single choice on a menu. If a menu item is not available, it appears dimmed.

message (1) A communication Instructor sends to an object to indicate that an event has occurred. (2) A warning or other information displayed by Windows or Instructor that alerts a user to a problem and possible corrections.

metafile Sometimes called a vector graphic. A file that defines an image as a series of instructions instead of as a collection of pixels (as in a bitmap or paint object). A metafile is exported from a Windows drawing program and, when imported into Instructor, becomes a picture object. *See also* picture object.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

MIDI (Musical Instrument Digital Interface) A series of commands sent to a synthesizer or sound card that controls pitch, length, volume, and so on. MIDI is not recorded sound; rather, it is sound that is created digitally.

modal A way of displaying a window so the user cannot perform any other action outside the window before closing it. For example, most dialog boxes are modal; until you close the dialog box, you cannot return to the program that displayed it. *Compare* nonmodal.

MP3 A file extension for the MPEG audio level 3 file format. MP3 files are small audio files that can be transferred across the Internet easily. The Universal Media Player in the Media Players category of the Catalog can play MP3 files.

multimedia Information in different formats—text, graphics, sound, video, and animation—that you can use in computer-based applications. See also animation, digital video, media file, MP3, MIDI, wave audio.

name One way of identifying an object. An object's name is a property of the object and is used when referring to the object. It is a good practice to assign a unique name to each object in your book, especially those objects with interactive functionality. Only named objects may be referenced in an action sequence. *Compare* ID number.

Neuron A plug-in that works with Netscape Navigator and Microsoft Internet Explorer that allows you to run native ToolBook II TBK files on the Internet. Neuron does everything necessary to run books remotely, including transferring clip-based media, system books, and DLLs from the Web server to your computer.

no-drop image The cursor, icon, or bitmap resource displayed during a drag-and-drop operation when the pointer is over an object that cannot be the destination object. *See also* destination object, drag and drop.

nonmodal A way of displaying a window so the user can leave it to activate another window. *Compare* modal.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

nonpersistent property A property that does not retain its value between Instructor sessions or instances. For viewers, nonpersistent properties exist only when a viewer is open and are not saved when a viewer is closed. *See also* viewer.

notification message A message sent after Instructor has finished an action that affects the status of an object. For example, notification messages are sent after an object is created, moved, or cleared, or when Instructor is idle.

notify handler A handler that is alerted automatically when a given message reaches the page. You can create notify handlers that are called before or after the page handles the message. Notify handlers are often used to create self-contained objects that can be copied into applications without having to change the scripts. *See also* handler.

object All of the visual elements of your pages (including buttons, fields, graphics, viewers, pages, and backgrounds). Any object can have a script or action sequence that defines its behavior and all objects have properties.

object hierarchy The order in which messages are passed from object to object. For example, a message sent to a graphic object that the graphic object's script does not handle is passed to the page, then to the background, then to the book, then to any system book, then to Instructor. *See also* message.

object linking and embedding (OLE) A Windows technology that enables you to create an object in one application and then incorporate it into another application. An object that is used this way is called an OLE object. *See also* client application, OLE container object, server application.

object-oriented programming A programming method in which programs are constructed out of self-contained collections of data and routines, referred to as objects.

offscreen image An image that Instructor first draws in memory and then displays on the screen. *Compare* draw direct.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

OLE container object An object that contains a reference to a linked object or a copy of an embedded object. *See also* client application, object linking and embedding, server application.

OpenScript The ToolBook II object-oriented programming language, used primarily in ToolBook II Instructor.

operand A value on which an operator performs an action. For example, in the expression a < b, the operands are a and b, and the operator is <. *See also* expression, operator.

operator In an expression, a symbol or word that causes Instructor to perform an action on operands, resulting in another value. For example, in the expression a < b, the operands are a and b, and the operator is <. *See also* operand.

orientation A setting that determines whether pages in a book are presented in portrait mode (taller than they are wide) or landscape mode (wider than they are tall).

overlay video A means of displaying analog video on computer monitors. When overlay video is played, Instructor designates the area of the screen where the video is to play. The video signal is then routed through and intercepted by the videodisc controller card, which replaces the designated area with data from the videodisc. *See also* analog, controller card.

page The object that is the basic unit of a book. Objects placed on a page are displayed on top of a background.

page number The relative order of a page among all pages in a book.

page size The dimensions of the pages in a book.

page units A measurement used to specify an object's bounds and location relative to the upper-left corner of the page. There are 1440 page units per inch, or 57 per millimeter. Page units do not vary in size with the resolution of the display device. *Compare* pixel.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

paint object An Instructor object created by importing a bitmap into Instructor or by pasting a bitmap from the Clipboard. When you drag the handles of a paint object to make it smaller, the image is cropped. *See also* bitmap, crop. *Compare* draw object, picture object.

paint-type graphic *See* bitmap.

palette (1) Predefined set of colors that can be imported and edited in Instructor. (2) Instructor's collection of tools used for drawing and selecting objects and specifying color, line ends and style, pattern, or polygon shapes.

palette colors The colors remaining in a color palette after Instructor has allocated its 16 or 32 standard colors and, optionally, another 64 custom colors. For example, if solidColorsEnabled is true on a 256-color system, Instructor makes 160 palette colors available.

palette flash Also known as palette shift. A change in the colors of graphics that occurs when you display a graphic with a color palette different from the one currently in use.

Palette Optimizer A utility you can use to adjust the colors used in imported graphics so that they achieve a close match to the colors that are available in the Instructor color palette. *See also* palette.

palette resource A color palette that you import for use in a book. When you import a color palette, all imported graphics in the book use colors from that palette. *See also* color palette.

parameter In OpenScript programming, the part of a statement that indicates which object or data a command is to act on, the data a function uses in the operation it performs, or the data passed with a message from one handler to another. Also, a value defined for a shared action sequence or ActiveX method that supplies information used in the execution of that particular shared action sequence or ActiveX method. *See also* function, handler, statement, shared action sequence.

parent object Any group, page, background, book, or system book that contains and is directly above another object in the object hierarchy. *See also* object hierarchy.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

parent window A window that owns another window. *Compare* child window, pop-up window.

path The course that an object follows as it moves during an animation. You create a path in the Animation Editor by clicking to create vertices that define the path. By setting the number of steps per second, the total duration of the animation, and the animation rate, you can control how quickly and smoothly the object moves along a path. The path is stored as a list of coordinates for each step of the animation.

pattern A repeating arrangement of pixels used to fill a background or draw object.

pattern palette A palette available at Author level from which you can assign a pattern to a background or any enclosed object. Patterns use the stroke and fill colors specified in the color tray. *See also* fill, stroke.

picture object Also called a vector or draw-type graphic. A graphic imported or pasted from the Clipboard that was created in a draw program. Picture objects can be resized in Instructor. *Compare* draw object, paint object.

pixel Short for picture element. The smallest element that display or print hardware and software can manipulate in creating letters, numbers, or graphics. Pixel size is dependent on the resolution of the display device. Screen units are measured in pixels. *Compare* page units.

plug-in A software program that adds functionality to a Web browser (plays sounds or digital video, or runs specific types of files over the Internet). A plug-in is stored on your computer and will open automatically when you request your browser to perform a function enabled by the plug-in.

point size A typographical unit of measurement for font size. One point is equal to $^{1}/_{72}$ inch.

pointer A symbol that shows the location of the cursor onscreen and that changes shape to indicate special locations and processes. For example, the pointer can be a selection tool, the Windows hourglass, or an I-beam.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

polygon An enclosed object with equal angles and a varying number of sides of equal length. *Compare* irregular polygon.

pop-up menu A menu that can appear anywhere onscreen within its own window.

pop-up window A window that generally appears in front of its parent window and minimizes when the parent window is minimized. A pop-up window can be dragged outside its parent's boundaries. Floating palettes are usually pop-up windows. All windows are either child windows or pop-up windows. *See also* child window, parent window.

precedence The order in which operators are evaluated. Operators with a higher precedence are evaluated before operators with a lower precedence. For example, the multiplication operator (*) has a higher precedence than the addition operator (+). *See also* operator.

Print Preview window The window that previews what a page will look like when it is printed.

printer system properties Instructor properties that affect how reports or pages are printed, such as the property printBorders.

properties Attributes of an object that define its appearance and behavior, such as the object's name, color, and whether the object can be dragged.

Properties dialog box A window in which you can view and edit an object's properties.

pushbutton A button style composed of a three-dimensional rectangle and a button caption that appears to move when clicked. Often used to trigger an action, such as navigating from one page to another.

query message A message sent to an object to determine its state or the value of a particular property. For example, during a drag-and-drop operation Instructor sends an allowDrop query message to each object under the cursor to determine whether the object will accept a drop. *See also* message.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

question object A Catalog object with preprogrammed functionality that allows you to construct different types of questions. Question objects can manage the user's response and score, and can also deliver feedback.

Quick Specialist A streamlined version of the Full Specialist with a select offering of settings. The Full Specialist, the Quick Specialist, and the Lesson Design Specialist guide you through the process of building a new Instructor book. *See also* Full Specialist, Lesson Design Specialist.

radio button A button style composed of a circle and a button caption. The circle toggles a dot on and off when a user clicks the active area of the button. Often used for a group of alternative items from which only one can be selected.

Reader level One of the two working levels in Instructor. At Reader level, a user can move between pages and books, click hotwords and buttons, make selections from combo boxes, and type text in editable fields. Applications are used and tested at Reader level. *Compare* Author level.

realize In 256-color systems, to activate a particular list of colors (a color palette) and make it the system palette. For example, when you display a bitmap, Windows realizes the palette associated with that bitmap.

record A group of related data fields in a database file. For example, the name, address, and phone number fields for one listing in an address book are a record.

record field A text field positioned on the background of a page that can contain different text on each page. *See also* background, text field.

red, green, and blue values (RGB) A means of specifying a color according to the amount of red, green, and blue in the color. *Compare* hue, lightness, and saturation (HLS).

reshape handles Handles that appear on a selected draw object when you choose Reshape from the Draw menu that you can use to reshape an object. Curves, angled lines, arcs, pies, and regular and irregular polygons can be reshaped.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

resources Interface elements (including graphics, cursors, fonts, icons, menu bars, palettes, and shared scripts) that can be imported and shared in one or more Instructor applications.

Rich Text Format (RTF) A text format that includes information about character and paragraph formatting for use in transferring formatted text between applications. Instructor preserves most character formatting and some paragraph formatting of text imported from any application that supports RTF.

right-click menu The menu that appears when you right-click an object.

rollover graphic The graphic that displays when the mouse pointer pauses over a button. In Instructor, you can assign a rollover graphic to a button in the Properties for Button dialog box by selecting the option labeled Use checked graphic as rollover graphic.

RTF *See* Rich Text Format.

ruler The horizontal or vertical scale used to measure and position objects on the page. A hairline on each ruler shows the current location of the pointer. *See also* grid.

run-time A version of Instructor that allows a user to run, but not modify, a book. A run-time version of Instructor is provided on the Instructor program disc.

sampling rate A measure of how frequently an event is tested, or sampled, to convert it from analog to digital format.

scope The specific handlers in which a variable's value is available. For example, the scope of a local variable is the current handler only. *See also* handler, variable.

script A series of statements written in OpenScript that define Instructor's behavior. Scripts are object properties and are divided into parts called handlers. Handlers define responses to specific events. When an event occurs, such as when a user clicks an object, Instructor runs the corresponding handler in the object's script. *See also* handler, statement.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

script editor The window in which you can create and edit an object's script.

scroll bars A bar along the right side and/or bottom of some windows and fields used to view all the contents of a window with more information than can be shown on screen at one time.

segment One leg of an animation path along which an object moves. When you draw an animation path, you click to create a new vertex. Instructor then draws a segment from the end of the existing path to the new vertex. By default, segments are straight lines, but they can be converted to curves.

select To identify the target for the next action. Selected text is highlighted. Selected objects display handles. When a page is selected, the page selection indicator is highlighted in the status bar.

selection The text, object, or page that is selected. An object selected at Author level becomes the target of commands you choose from the Instructor menus. Move or resize the selection using its handles. When text is selected, it can be edited or formatted.

selection rectangle A dotted box that appears when you drag the pointer across a page. When the mouse button is released, Instructor selects all objects completely enclosed by the selection rectangle.

server A computer connected to a network or the Internet (or both) that contains documents available via the network or the Internet.

server application An application that creates objects that can be linked or embedded in a client application. Instructor is a client application that can receive OLE information, such as graphics or text, from server applications. *See also* client application, object linking and embedding, OLE container object.

session One or more instances of any Windows application. *See also* instance.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

shared action sequence An action sequence that runs only when executed as part of another action sequence. You can create a shared action sequence that performs an operation and returns a value to the action sequence that calls it. You can define parameters the shared action sequence uses to complete any operation it performs. *See also* action sequence, parameter.

shared script OpenScript code that can be shared among Instructor objects. Once defined, the shared script is stored as a resource in the book and can be applied to an object as needed.

silent dragging An option for drag-and-drop behavior that specifies that the source object can be dropped anywhere, even if the destination object does not normally allow a drop. Silent dragging is used to create drag-and-drop behavior for applications (such as tool palettes or clip art books) that might be used in conjunction with applications not configured to receive drop objects.

solid color A color that appears on a color display device when all pixels are the same color. *See also* pixel.

sorting The process of placing pages in order based on the contents of specified record fields.

sound An audio CD, MIDI, MP3 or wave audio sound that can be played in an application. *See also* MIDI, MP3, wave audio.

source book The book from which pages are imported into another book.

source object In a drag-and-drop operation, the object that is clicked to begin the operation and whose drag image is dragged. *See also* drag image, drag and drop.

special value A type of variable with a system-defined value. Special values available in the Actions Editor include pi, true, and false.

spooler A temporary storage location for information that is being printed. Instructor gathers the information to be printed in a spooler and then sends the contents of the spooler to the designated printer. In OpenScript, you must use a spooler control structure to print.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

stack A list organized so that the last data item added to the list is the first item used.

standard colors The 16 or 32 colors that are always available to Instructor applications.

startup screen A screen that appears for a few seconds upon startup of an application and then disappears. A startup screen is often used as the application's introductory screen, containing information such as the application's title, release number, and the name of the person to whom the software is registered.

startup system properties Instructor properties that specify how Instructor is initialized when you first start it. For example, the system property startupBook specifies the name of the book that opens automatically when a user runs Instructor.

statement A single complete instruction in a script or action sequence.

static object An OLE object that was embedded or linked but whose link was severed. A static object has no link to its server application. *See also* object linking and embedding, server application.

status bar The bar along the bottom of the Instructor window that shows mouse position, name of the object under the cursor, information about the chosen menu command, and the page selection indicator.

string Any combination of contiguous characters.

stroke The line element of an object. For fields and buttons, stroke color is applied to text and borders. For draw objects, stroke color is applied to borders and the stroke elements of any pattern used. For paint objects and picture objects, stroke color is applied to the border. *Compare* fill.

style (1) An object's visual attributes, such as its border type, line style, or pattern. (2) The bold, italic, underline, and strikeout text attributes that can be set using the Character dialog box.

submenu A menu item that displays a cascading menu when clicked. Submenus are indicated by small arrows to the right of menu items.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

syntax The rules for writing script statements in OpenScript or for creating expressions in the Build Expression dialog box of the Actions Editor.

system book A book whose book script is shared by other books. System books are useful for defining general scripts for repeated use by other books. For example, you can put all of your user-defined functions and user properties into a system book so that they are available for any applications you create. *See also* user property, user-defined function.

system palette In a 256-color system, the list of colors currently used by the video display device to display graphics. The 256 colors are drawn from the 16 million colors available to Windows.

system properties Property that define default behavior for the entire Instructor system.

system variable A variable that maintains its value for the duration of an Instructor instance or session. Any handler that declares the system variable can access and change its value during a session. *See also* handler, local variable, global variable.

tab (1) A character in text that Instructor uses to align text in fields. Tabs are set using the Paragraph dialog box. To type a tab character in a field, press CTRL+TAB. See also field. (2) The TAB key. Pressing TAB moves the focus between buttons, editable fields, and combo boxes. *See also* tab order.

tab order The order in which Instructor moves the focus between buttons and editable fields when a user presses the TAB key. The tab order can be modified by changing the layer order of objects. *See also* focus, layer, tab.

target A system property that specifies which object received the current message. You cannot set this property. *See also* system properties.

target book The book into which pages are imported from another book.

target window The viewer in which Instructor executes commands and searches for objects. The commands can affect any page, background, or object displayed in the viewer, or the viewer itself. *See also* viewer.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

template A prebuilt book that you customize by adding content.

text field An Instructor object that holds text. A text field has properties that determine its appearance and behavior.

textline One data element out of a text string separated by carriage return/linefeeds (CRLF). The textline does not include the CRLF, which is used only as a delimiter. *See also* carriage return/linefeed, delimiter.

tile A property of a window that contains a caption or the file name.

title bar The bar at the top of a window that contains a caption or the file name.

ToolBook II Web Specialist A wizard that prepares your ToolBook II application for Internet distribution. The Web Specialist can automatically export your application to DHTML. You can open the Web Specialist by selecting Publish to Web from the File menu. *See also* application, DHTML.

tool palette The tools that appear at Author level for creating and working with objects.

toolbar The bar containing buttons that are shortcuts for frequently used menu commands. Displayed by default at the top of the Instructor main window at Author level.

trace To go through the execution of a script in the Debugger one statement or keyword at a time to locate an error. *See also* Debugger, script.

transition A special effect, such as a fade-out, that occurs as a user navigates from one page to the next.

transparent A property that determines whether an object is displayed so that objects on layers behind the object can be seen through the object. *See also* layer.

ungroup To separate grouped objects into individual objects. Ungrouping discards all the group's properties and its script but does not affect individual object's scripts and properties. *See also* group.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Uniform Resource Locator (URL) An address used to locate Web pages on Web servers. Much like your area code and telephone number identify your region and then your location, the URL of a Web page identifies a computer and then a specific document.

unique identifier A method of identifying an object in any book. Includes the object type name (such as button, page, field, and so on), object ID number, and the unique name of the page or background on which the object is located.

URL *See* Uniform Resource Locator.

user property A property that a developer creates for an object beyond those which are built into the Instructor object, and for which the developer can get and set values as with any other property. All objects in Instructor can have user properties. For a list of user properties defined for an object, get the value of the object's userProperties property.

user-defined function A function in OpenScript that is defined by a handler written by the developer, in contrast to a function that is built into OpenScript. *See also* function, handler.

user-defined message Any message that is not one of the built-in messages that Instructor sends in response to an event. The developer must provide a corresponding handler for all user-defined messages. Instructor's default response to user-defined messages that are not handled is to display the Execution Suspended message. *See also*
Execution Suspended message, User event. Compare built-in message.

User event A generic event that you can send using the Send User Event action in the Actions Editor. When you send a User event, you can send data (a literal value, variable, or expression) in a single parameter. *See also* event.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

value A piece of data such as text, a number, or a logical value. For example, in the following statements, Joe, 3456, and true are all values:

```
text of field "name" = "Joe"        --Text value
svQuantity = 3456                   --Numerical value
visible of selection = true         --Logical value
```

See also container.

variable A named container that holds data in action sequences and OpenScript programs. *See also* global variable, local variable, and system variable.

variables window The window that displays the current values for local variables and system variables when you work in the Debugger. *See also* Debugger, local variable, system variable.

vector graphics *See* draw-type graphic.

vertex, vertices The points defining the beginning and end of a segment in a path animation. When you draw a path animation, you click to position a vertex, and Instructor draws a segment between the vertex at the end of the path and the new vertex.

video overlay *See* overlay video.

viewer A window that you can create in Instructor that displays pages from any book. Viewers are composed of a frame window, a client area, and a client window. Viewers can be used to create dialog boxes, tool palettes, status bars, pop-up windows, and application startup screens. *See also* client area, client window, frame window, pop-up window.

visual media clip Clips created from animation, video, photo CDs, or still images. *See also* clip.

wave audio Sound recorded in a digital format and stored as a file with the file extension .wav.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Web document An HTML file on a Web server. A Web document becomes a Web page when it is viewed through a browser. *See also* Web page.

Web page A document that can be viewed and interacted with over the Internet. Web pages can contain text, graphics, and links to other Web pages. Web pages make up the content of the World Wide Web. *See also* home page, Web document, World Wide Web.

Web Specialist A wizard that helps you to prepare your Instructor application in a format that can be distributed on the Internet.

wildcard A question mark (?) used to represent a single character, or an asterisk (*) used to indicate multiple characters in a file name or search text.

window The fundamental interface object of Microsoft Windows. *Compare* instance.

wizard An interactive utility that guides a user through a process step by step. The Full Specialist is an example of an Instructor wizard.

wordwrap An option that automatically places text on the next line when you type past the right margin in a field. Text does not wrap in button captions. *See also* field.

World Wide Web (WWW) A part of the Internet with the capacity to handle multimedia (text, graphics, video, and sound) information. Documents, called Web pages, that are stored on computers around the world make up the content of the World Wide Web.

WWW *See* World Wide Web.

zoom To magnify the view of a page using the magnify tool from the tool palette.



Index

A

access keys 566–567
actions
  disabling 400
  enabling 400
  expressions in 383–385
  inserting in action sequences 377
  overview 376, 378
  setting properties 381
  validating 382, 397
  See also action sequences
Actions Editor
  adding actions to ActiveX controls 597–600
  adding actions to media file time markers 340–343
  adding navigation behavior to objects 122–123
  as Author-level tool 59, 72–74
  creating action sequences 370–388
  hiding and showing objects 316–317
  illustrated 369
  opening 368–369
  overview 49, 364
  programming button behavior 258
  using to play animations 325
  using to specify feedback 310
  See also action sequences
Actions Editor toolbar 371
Actions Editor Walkthrough book 405, 406
action sequences
  compared with Catalog objects 367
  compared with OpenScript 367
  conditions example 407–413
  creating using Actions Editor 370–388
  and DHTML 660
  examples overview 404–406
  exporting 400–402
  illustrated 369
  importing 400–402
  managing 396–402
  object properties example 421–426
  overview 364–366
  shared 366, 388–396, 402, 426–440
  shared actions example 426–440
  troubleshooting 441
  variables example 414–421
  viewing 398–399
  when to create 366–367
actions palette 372
action triggers
  defined 132
  for hide-and-show effects 314–315
  for playing animations 323–324
activate command 528, 529

ActiveX
  adding controls to books 588–589
  calling control methods 594–595
  compared with OLE 51
  creating action sequences for controls 597–600
  drawing controls on pages 589
  executing methods in action sequences 599
  handling events 595–596, 600
  Instructor support for 50
  overview 586, 587, 662
  removing controls from books 590
  setting control properties 592–593, 597–598
  troubleshooting controls 601
  writing OpenScript code for controls 591–596
  See also ToolBook II Neuron
ActiveX Data Objects (ADO) 50, 52, 603
add menu command 563
add menuitem command 563
ADO. *See* ActiveX Data Objects (ADO)
AICC support 667
aliases 551, 556, 567–569
aligning objects 155–157
allowAuthorActivate property 608
allowReaderctivate property 608
alwaysOnTop property 524
alwaysReader property 524
angled lines, drawing 146
angled line tool 145
animated GIF placeholder 134
Animation Editor. *See* ToolBook II Animation Editor
animations
  cel-based 318, 321–322
  modifying 327–329
  path-based 318, 320–321
  playing at Reader level 322–327
  selecting to edit 328
  setting options 329
  testing 321, 322
  See also clips
ANSI characters, adding to text 216–217
applications. *See* online learning applications; Windows programs
arc tool 145
arithmetic expressions 383
arrange-objects object 297
arrays in Actions Editor 387–388
arrow keys 87
ASCII files 470
audio files 335
Author level
  Actions Editor 59, 72–74
  browsers 59, 74–78
  Catalog feature 67–68, 126–139
  color tray 58, 66
  compared with Reader level 42, 84–85
  Extended Properties dialog box 59, 70, 71–72, 180, 186–188
  grid 58, 59, 80–81
  illustrated 57, 58

Author level, *continued*
list of interface elements 59
menu bar 58, 59, 60
navigating books 119–123
overview 42, 43
palettes 58, 59, 65–67
Properties dialog box 59, 70–71
right-click menus 58, 59, 81–84
rulers 58, 59, 79–80
status bar 58, 59, 63–64
switching to Reader level 85
toolbar 58, 59, 60–63
tool palettes 65–67, 145
authorStatusBar property 64
autoClose property 524
Automation
calling object methods 594–595
creating objects 590–591
handling events 595–596
Instructor support for 50, 51
overview 586–587
setting object properties 592–593
troubleshooting 602
writing OpenScript code for objects 591–596
AutoPackager
copying files 650
gathering files 649
overview 647
packaging application files 649–650
as tool for preparing applications for distribution 642–644, 647
using to package books 648–651
autoShow property 524
autoSize property 515

B

backdrops 114
backgrounds
adding backdrops to 114
applying colors to 113–114
applying patterns to 113–114
changing 103
compared with foreground 92
displaying right-click menus 83
illustrated 93
new compared with existing 103
optimizing 612–613
overview 92–93, 113
and pages 44, 45
removing 112
removing backdrops 114
setting page size 115–117
setting properties using OpenScript 114
shared 112
storing offscreen images 625–629
switching to 92, 103
See also layers
backup copies, saving 641
Behavior tab, Properties dialog box 182
benchmarking performance 630–631
bitmaps
creating compressed images 625–626
creating resource type 449
creating transparent areas 261–262
device dependence 627
displaying efficiently 625–629
getting resource information 463
paint objects as 228
printing 675

bitmaps, *continued*
removing compressed images 628–629
removing references to resources 467
as resources 447, 466
storing compressed images 627
updating compressed images 628
See also graphics
blank books, starting 101
BMP files 228, 470
book as metaphor 44
book palettes 239, 240, 241, 242, 457–458
books
adding graphics to 229–233
adding hyperlinks to pages 148–150
adding pages to 103–105
adding resources to 452–454
assigning color palettes to 457–458
basing on templates 99–100
changing system cursor 461
compressing 640–641
creating 98–102
directory structure for 37–38, 636–637
displaying right-click menus 83
importing 486–488
list of distribution options 39–41, 643, 658, 660
navigating 86–87, 119–123
optimizing 124, 612
overview 44, 92
packaging using AutoPackager 648–651
printing pages 672–675
protecting with passwords 118–119
publishing to Web 662–666
read-only access to 652
books, *continued*
resetting 641–642
saving as templates 97–98
saving with new name 640–641
setting properties 118–119
specifying page size 115–117
starting from scratch 101
starting using Book Specialists 101–102
testing 124, 653
unused resources 466–468, 639
as viewer parent objects 502
See also Book Specialists; online learning applications
Book Specialists
Full Specialist 102
Lesson Design Specialist 102
Quick Specialist 102
starting new book 101–102
borderless field tool 145
borders
changing color 235
printing page borders 686
styles for buttons 254
styles for fields 199
borderstyle property 253
bounds property 234
Bounds tab, Properties dialog box 182
browsers, Instructor
as Author-level tool 59, 74–78
defined 59, 74
Handler Browser 78
Object Browser 76
opening 74
Page Browser 77
Property Browser 75, 188–190

Build Expressions dialog box 384
buttonClick handler 502, 521, 522
buttons
  action buttons 132
  Actions Editor toolbar 371
  adding from Catalog 249
  adding graphics to 258–260
  adding hyperlinks to 256
  adjusting layout 262
  assigning graphics to 454–456
  Author-level toolbar 61–62
  border styles 254
  compared with hotwords 263
  creating using OpenScript 250–251
  creating using tool palette 250
  and DHTML 660
  formatting caption text 255
  navigation buttons 135
  overview 246–248
  programming behavior 258
  scripting 257
  setting properties 249, 251–254
  uses for 246

C

cache files 646
captionPosition property 253
caption property 253
captions
  adjusting button layout 262
  formatting button text 255
  setting viewer styles 511
cascading menus 552
Catalog
  adding and changing backdrops 114
  adding graphics from 232
  adding objects to 138
  as Author-level tool 59, 67–68
  browsing 127
  compared with action sequences 367
  customizing 137–140
  deleting objects from 139
  displaying 68, 127
  drag-and-drop authoring 127–130
  draw objects illustrated 143
  and extended properties 129, 130
  fields in 192–193
  finding objects 130–131
  illustrated 68, 126
  modifying categories 137
  naming objects 138
  navigation objects 120, 135
  overview 35–36, 126
  players in 332, 333–334
  removing objects from 139
  searching for objects 130–131
  types of objects 132–136
  using 67–68
  viewing objects full size 131
  viewing question objects 293
CDMediaPath property 356, 359
CD-ROMs
  and cache files 646
  distributing applications via 41, 643
  optimizing books for 645
  organizing files on 355–356

CD-ROMs, *continued*
packaging applications using AutoPackager 648–651
preparing files for 645
testing installations 653
cel animations 318, 321–322
centering objects 154
CGM files 233
check box tools 145, 250
checking spelling 638
check menuItem command 564, 571
child windows
compared with pop-up windows 500
tiling 498, 499
viewers as 497–498
chromakey property 238
clear message 112
click2learn.com
contacting technical support 30
family of products 34
learning management solutions 668
See also ToolBook II Instructor
click2learn.com Web site
Developer Corner 28
information resources 29–30
clientHandle property 534
clientSize property 516
client windows 500, 501
Clipboard, copying and pasting graphics 230
clip command 350
Clip Manager 347, 348, 351–352
clips
compared with media files 343, 347
copying 351
creating 347–350
clips, *continued*
creating libraries 352
editing 351
Instructor searches 356–357
making paths independent 354–356
managing 350–352
overview 343, 347
playing 351
positioning in stages 346
removing 350, 351
replacing 351
sizing in stages 346
and transition effects 347
See also media players; stages
closeWindow message 502
Coach. *See* ToolBook II Coach
color
applying to backgrounds 113–114
applying to hotwords 268
applying to objects 162–164
applying to text in fields 211–212
creating custom colors 162
setting for viewers 512
Windows system colors 163–164
Color dialog box 163
color palettes
applying to graphics 241
assigning book palettes 242
assigning to books 457–458
creating book palettes 239, 240, 457–458
hiding/showing 66–67
moving 67
overview 65
palette flash 239, 457
Palette Optimizer 239, 240

color palettes, *continued*
priority among 457
removing references to resources 467
using with viewers 522–523
variations among 238–239, 457
color tray, illustrated 58, 66
column reports 675–676, 682, 685
combo boxes
adding from Catalog 281
adding text to 284–285
creating 281–282
determining user selection 286, 288–290
Internet compatibility 274
overview 273
setting properties 283–284
uses for 273
combo box tool 145
compressed bitmap images 625–629
compressing books 640–641
computer screens. *See* displays
conditions, action sequence 376, 379–380, 407–413
constraining objects 159–160
controls. *See* ActiveX; buttons
convertPicture message 237
coordinates, for positioning viewers 515–520
copying
graphics from Clipboard 230
objects 151, 152
resources 451, 453
text 204, 206–208
viewers 507
See also exporting; importing
copy Resource command 453
courseware. *See* online learning applications
createHotword message 265
cropping graphics 233–234
currentPage property 525
cursors
changing system cursor 461
creating resource type 449
for dragging 458–459
exporting resources 465
getting resource information 463
no-drop 458–459
removing references to resources 467
curves, drawing 147
curve tool 145
custom colors 162
cut message 112

D

data types 620–621
DDE. *See* Dynamic Data Exchange (DDE)
defaultClientSize property 516
delay button 132
delayed feedback
compared with immediate feedback 306–307
specifying 309–310
deleting. *See* removing
delimited-field files
compared with fixed-field files 471
exporting record field data to 482–484
importing 476–478
delivering online learning applications. *See* distribution options, online learning applications

Developer Corner, click2learn.com Web site 28
DHTML. *See* Dynamic HTML (DHTML)
dialog boxes, creating viewers to display 535–541
directories, organizing files 37–38, 636–642
disable commands 564, 572–573
disk space requirements 22
displaying. *See* hiding/showing
displays
  differences among 627
  optimizing for draw objects 616
  and stored bitmaps 627
  See also page size
distribution file list 29
distribution options, online learning applications 39–41, 643, 658–662
dithering 241
DLLs. *See* dynamic-link libraries (DLLs)
documentation, Instructor 26–28
Drag & Drop tab, Properties dialog box 182
dragging and dropping
  Catalog objects 127–130
  to move objects 153
dragImage property 459
drag images 458–459
drag-objects object 298
draw command 196, 251, 282, 345
drawDirect property 625
draw objects
  creating 144–147
  illustrated 143
  optimizing display 616
  See also graphics; objects
Draw tab, Properties dialog box 182
dropDownItems property 285
drop-down menus 551–552
drop-targets object 299
duplicating
  objects 151, 152
  resources 451
  See also copying
Dynamic Data Exchange (DDE), Instructor support for 50, 52
Dynamic HTML (DHTML)
  exporting applications to 664
  as Internet distribution option 658, 659–660
  and OpenScript 50
  optimizing applications for 610
  preparing files for release 643, 644
dynamic-link libraries (DLLs), extending OpenScript with 52

E

e-Learning applications. *See* online learning applications
ellipse tool 145
embedded OLE objects 604, 605–606
enable commands 564, 572–573
enabled property 253
enterApplication handler 521
enterMenu message 581–582
EPS files 233
events
  creating action sequences for 372–376
  menu message handling 576–582
excludeTab property 253

exiting Instructor 25
export command 465, 482, 484
exporting
  action sequences 400–402
  applications to DHTML 664
  graphics 485
  native ToolBook II books for Web 665–666
  overview 470–471
  record field data 479–484
  resources 465
  text 471
  text field data 471, 472–473
  when to use 470
expressions 383–385
extended properties
  and Catalog objects 129, 130
  overview 48, 180, 186
  setting for groups 187
  setting for objects 187
  setting for question objects 295–296
Extended Properties dialog box 59, 70, 71–72, 180, 186–188
extensions, unused, removing 640

F

feedback
  immediate compared with delayed 306–307
  specifying 304–310
  types of 305
Feedback field 133
fields. *See* record fields; text fields
field tool 145
fieldType property 276
file formats
  graphics 229, 233, 470
  for importing and exporting 470
  Internet-based 229, 332
file placeholder 134
files
  cache files 646
  copying in AutoPackager 650
  organizing for book distribution 37–38, 353–356, 636–642
  reducing size 447, 466, 640–641
  See also books
fillColor property 114, 212, 238
filling objects with patterns 165
fill-in-the-blank objects 136, 300
filters, graphics 231
finding
  Catalog objects 130–131
  text in fields 218–219
fixed-field files
  compared with delimited-field files 471
  exporting record field data to 480–482
  importing 474–476
Flash Player 335
flipping objects 154–155
floating tool palette, creating 541–543
focusWindow property 529
focus windows 527, 528–529
folders. *See* directories, organizing files
fontFace property 681
fonts, creating resource type 449
fontStyle property 681
Font tab, Properties dialog box 182

footers, adding to printed pages 687–689
foreground. *See* backgrounds; layers; pages, book
formatting
  field character styles 210–211
  field paragraph styles 213–214
  report text 681–683
  text in fields 209–214
frames, for stages 346
frame windows
  defined 500
  illustrated 501
  windowHandle property 534
Full Specialist 102
functions
  for coordinate conversion 519
  in expressions 384

G

GIF files 229, 470
global variables 386, 392
Go menu 87
graphics
  adding from Catalog 232
  adding to books 229–233
  adding to buttons 258–260
  applying color palettes 241
  assigning to buttons 454–456
  changing border color 235
  changing line style 235
  converting picture objects to paint objects 236–237
  creating transparent areas 261–262
graphics, *continued*
  cropping 233–234
  device-dependence 627
  dragging and dropping 129
  exporting 485
  file formats 229, 233, 470
  filters 231
  identifying types 228
  importing 231, 484–485
  inline, inserting 214–215
  Internet-compatible formats 229, 332
  modifying 233–238
  optimizing display 616
  overview 228
  paint objects compared with picture objects 228
  pasting from Clipboard 230
  removing references to resources 467
  as resources 232, 447, 466
  storing files 233
  transparent areas 114, 237–238
  vector 228, 233
  See also bitmaps; clips; draw objects
grid
  as alignment alternative 155
  changing settings 81
  changing unit of measurement 81
  defined 59, 80
  hiding/showing 81
  illustrated 58, 80
  snapping objects to 80
  using 80
group reports 676, 685

groups
adding objects 175–176
creating 172
deleting objects 176
deselecting 173
editing 173
editing objects within 174
moving objects 174
nesting 178
optimizing 617–618
overview 172
selecting 173
selecting objects within 174
setting extended properties 187
setting properties 177
ungrouping objects 178

H

Handler Browser 78
handlers
for menu-event messages 576–582
for optimizing OpenScript programs 619–620, 623–624
hardware requirements 22, 23
HDMediaPath property 356, 359
headers, adding to printed pages 687–689
Help, online
accessing Help system 27
adding text to menus and menu items 569–570
text in status bar caption area 63
hidden message 502
hide-and-show effects 312–317
hide command 64, 67, 521
hide menuBar command 564, 573
hideOnDeactivation property 524
hideOnReader property 524
hiding/showing
grid 81
menu bars 573–574
objects at Reader level 312–314
palettes 66–67
rulers 80
status bars 64
toolbars 63
viewers 523–524
highlight property 253
hotwords
applying color to 268
changing to regular text 269
compared with buttons 263
creating 264–267
defining hyperlinks 264–265
deleting 269
editing 267–268
overview 263–264
and report formatting 682
setting properties 266–267
uses for 263
See also hyperlinks
hotwordStyle property 266–267
HTML launch page for Neuron book 666
HTML placeholder 134
Hyperlink dialog box 149, 150
hyperlinks
creating for buttons 256
creating for objects 148–150
destination options 149–150

hyperlinks, *continued*
and DHTML 660
in a graphic 135
overview 148
testing 653
types of 149–150
to Web sites 149, 150
See also hotwords

I

icons
assigning to viewers 512
changing 460
creating resource type 449
exporting resources 465
getting resource information 463
removing references to resources 467
image buffers, assigning to viewers 512–513, 613
images. *See* bitmaps; graphics
immediate feedback
compared with delayed feedback 306–307
specifying 307–308, 310
import command 107, 223, 453, 476, 478, 488, 490, 491
Import dialog box 106
importGraphic command 231
importing
action sequences 400–402
books 486–488
customizing process 490–491
delimited-field files 476–478
fixed-field files 474–476
graphics 231, 484–485
overview 470–471
importing, *continued*
pages from another book 105–107, 223, 488–490
pages with record fields 223
resources 453, 454
text 203
text files into record fields 471, 473–478
text files into text fields 471–472
when to use 470
Impulse utility 661, 672
Ingenium 34, 668, 670
in-line clip players 334
inline graphics, inserting into text 214–215
insert graphic command 215
inssertOLEObject command 606
installing Instructor 24
InstallShield Express program 651
InstallShield Uninstall program 654
Instructor. *See* ToolBook II Instructor
interactive online learning applications. *See* online learning applications
interface, Instructor
Actions Editor 59
browsers 59, 74–78
Catalog 67–68
Extended Properties dialog box 59
grid 58, 59, 80–81
menu bar 58, 59, 60
overview 56
palettes 58, 59, 65–67, 145
Properties dialog box 59, 70–71
right-click menus 58, 59, 81–84
rulers 58, 59, 79–80
status bar 58, 59, 63–64
toolbar 58, 59, 60–63

Internet
choosing distribution option 657–662
compatible graphics formats 229, 332
DHTML as book distribution option 643, 658, 659–660
exporting Instructor applications to Web 662–666
Neuron as book distribution option 642, 643, 658, 661
overview of application distribution options 40, 643, 656–658
intranets, distributing applications via 40, 643
invert property 267
irregular polygon tool 145
isOpen property 522

J

Java placeholder 134
JPEG files 229, 470

K

keepMenuBar property 575
keyboard shortcuts for navigating text 205
keys, access 566–567

L

label button tool 145, 250
LANs
distributing applications via 41, 643, 644
packaging applications using AutoPackager 648–651
using Instructor on 652
layers
changing order 167–169
and focus 167, 169
illustrated 166
moving objects between 166–172
overview 166–167
planning order for Reader level tabbing 169–172
See also backgrounds
layout, changing in Print Preview 685
learning applications. *See* online learning applications
learning management systems 667–669
Lesson Design Specialist 102
libraries, clip, creating 352
lineCount property 282
line ends palette 65
line palette 65
lineStyle property 235
line styles, changing 235
line tool 145
links. *See* hyperlinks
list boxes
adding from Catalog 275
adding text to 279–281
creating 274–276
determining user selection 286–288
Internet compatibility 274
overview 272
setting properties 276–279
single-select compared with multi-select 278–279
uses for 272

local area networks (LANs)
distributing applications via 41, 643, 644
packaging applications using AutoPackager 648–651
using Instructor on 652
local variables 386, 392
locking fields 200
lockScreen property 526, 527
logical expressions 383
loops, action sequence 376, 380–381

M

Macromedia Flash Player 335
macros. *See* scripts
magnify tool 145
margins 683, 684
match-items objects 136, 300–301
media files
compared with clips 343, 347
compatible formats 333–334, 335
gathering using AutoPackager 649
organizing 353–356
search paths 356–359
selecting 336–337
testing 653
time markers in 337–343
See also clips; media players
media player buttons 135
media players
adding from Catalog 344
as Catalog objects 135
list of 333–334
overview 135, 332, 333, 343
and stages 344–347
media players, *continued*
video players 334
See also Universal Media Player
Menu Bar Editor 553–554, 556
See also menu bars; menus and menu items
menuBar property 561, 562
menu bars
assigning to viewers 511, 560–561
creating resource type 449
creating using Menu Bar Editor 449, 556–559
defined 59
and DHTML 660
getting resource information 463
hiding/showing 573–574
how menus work 555
illustrated 58, 60
Internet compatibility 551
modifying using OpenScript 562–583
overview 550–551
removing from viewers 562
removing references to resources 467
as resource type 550, 559–562
restoring to original state 574
tools for creating 553–554
using 60
See also menus and menu items
menuItemSelected message 555, 578–579
menus and menu items
adding access keys 566–567
adding check marks 571–572
adding Help text 569–570
adding separator bars 572
adding submenus 566

menus and menu items, *continued*
adding using Menu Bar Editor 556–559
adding using OpenScript 565–566
aliases for 556
cascading 552
creating using OpenScript 565–566
disabling 572–573
displaying items conditionally 581–582
drop-down menus 551–552
enabling 572–573
how they work 555
pop-up menus 553, 582–583
removing using OpenScript 574
renaming 570–571
submenus 552
types of 551–553
using OpenScript compared with Menu Bar Editor 553–554
See also right-click menus
messages
menu-event 576–582
and object hierarchy 501–502
and viewers 502
Microsoft Excel, bringing worksheets into Instructor applications 51
Microsoft Streaming Media files 335
MIDI files 335
Minimize icon, changing 460
mmSearchCD property 356
mmSearchHD property 356
mmSource property 356
modal viewers 523
monitors. *See* displays
mouse, status bar position indicator 63
moved message 502
moving
objects between layers 166–172
objects on pages 153
palettes 67
MP3 files 335
MPEG files 335
multimedia
Internet-compatible formats 332
overview 332
system requirements 22, 23
See also clips; media files
multiple-choice questions 136, 301

N

name property 345
naming Catalog objects 138
navigating books
adding actions 122–123
objects for 120, 135
optimizing 615
overview 119–120
restricting 506
skipping pages 121
testing 653
transition effects 121–122
using built-in tools 120
using status bars 63, 86–87
navigating text, using keyboard shortcuts 205
nesting groups 178
networks
distributing applications via 40–41, 643
using Instructor on 652
See also Internet; intranets

Neuron. *See* ToolBook II Neuron
New Background dialog box 104
newBackground message 105
New Book dialog box, bypassing 99
new clip command 351
New Page dialog box 104
newPage message 105
new viewer command 505
no-drop cursor 458–459
noDropImage property 459
nonmodal viewers 523
nonpersistent properties 508

O

Object Browser 76
object linking and embedding. *See* OLE objects
object properties 385, 421–426
objects
  adding navigation actions 122–123
  adding from Catalog 127–130, 142–144, 232
  adding to Catalog 138
  adding user properties to 190
  aligning 155–157
  animating 318–322
  applying color 162–164
  applying line styles 161
  and browsers 75, 76
  buttons 246–262
  Catalog types 132–136
  centering on pages 154
  changing layer order 167–169
  constraining shapes 159–160
  copying 151, 152
objects, *continued*
  creating 144–147
  cutting 151, 152
  deleting 153
  and DHTML 660
  dragging and dropping 127–130
  dragging to move 153
  drawing 144–147
  duplicating 151, 152
  filling with patterns 165
  finding in Catalog 130–131
  flipping 154–155
  grouping 172–178
  hiding/showing 312–314
  hierarchy of 501–502
  hotwords 263–269
  hyperlink capabilities 148–150
  making invisible 164
  making transparent 164
  moving between layers 166–172
  moving on pages 153
  naming 138
  for navigation 120, 135
  and Object browser 76
  optimizing display 616
  optimizing position 613–614
  overview 47–48, 142
  paint compared with picture 228
  pasting 151, 152
  placeholders as 134
  positioning 153–157
  and Property Browser 75, 188–190
  referencing in target window 531–532
  removing from Catalog 139
  reshaping 158–159

objects, *continued*
resizing 157–158
and right-click menus 83
rotating 154–155
selecting 147–148
setting properties 177, 183, 185–186
specifying line ends 161
spreading on pages 154
storing offscreen images 625–629
tab order 169–172
viewers as 502
viewing full size in Catalog 131
See also OLE objects; text fields; ToolBook II Instructor Catalog
OLE Automation. *See* Automation
OLE objects
adding to Instructor applications 51, 605–606
compared with ActiveX controls 51
creating and embedding 605–606
editing 606–607
Instructor support for 50, 51
overview 603–604
removing 608
setting properties 607–608
and versions of OLE technology 604
online Help. *See* Help, online
online learning applications
benchmarking performance 630–631
creating startup screens 547–548
directory structure 636–637
including other applications in 51
installing 653
interactive 37
online learning applications, *continued*
and learning management systems 667–669
navigating 86–87
optimizing 610–630
organizing files for distribution 37–38, 353–356, 636–642
overview 36–37
overview of distribution options 39–41, 643, 658–662
packaging using AutoPackager 648–651
planning 37–41
printing pages 672–675
publishing to Web 662–666
spelling checking 638
testing 124, 653
uninstalling 654
See also books
opening
Animation Editor 319
Book Specialists 102
ToolBook II Instructor Catalog 127
viewers 521
OpenScript
adding access keys to menus and menu items 566–567
adding aliases to menus and menu items 567–569
adding book pages 105
adding check marks to menu items 571–572
adding graphics to buttons 260, 456
adding headers and footers to printed pages 689

OpenScript, *continued*
adding Help text to menus and menu items 569–570
adding media search paths 359
adding menus and menu items 565–566
adding resources 453
adding separator bars to menu items 572
adding submenus 566
adding text to combo boxes 285
adding text to fields 202
adding text to list boxes 280
additional information about 26, 31
assigning color palettes to books 458
assigning menu bars to viewers 561
changing cursors 459
changing graphic border colors 235
changing graphic line styles 235
changing hotwords to regular text 269
changing Minimize icon 460
changing system cursor 461
closing target window 532
closing viewers 521
compared with action sequences 367
compared with Menu Bar Editor 553–554
compared with Visual Basic 591–592
converting picture objects to paint objects 237
copying and pasting text into fields 207–208
creating Automation objects 590–591
creating buttons 250–251
creating clips 350
creating combo boxes 282
creating custom viewers 505

OpenScript, *continued*
creating fields 196
creating hotwords 265
creating list boxes 276
creating stages 345
cropping graphics 234
customizing importing 490–491
determining combo box selection 288–290
determining list box selection 287–288
and DHTML 660
disabling menus and menu items 572–573
displaying menu items conditionally 581–582
displaying pages in viewers 524–525
displaying pop-up menus 582–583
and DLLs 52
enabling menus and menu items 572–573
enabling Reader-level right-click menus 84
exporting data from record fields to delimited-field text files 484
exporting data from record fields to fixed-field text files 482
exporting resources 465
formatting report text 682
formatting text in fields 210, 211, 212, 214
getting resource information 462
handling menu-event messages 576–582
hiding/showing menu bars 573–574
hiding/showing objects 317
hiding/showing palettes 67
hiding/showing status bar 64
hiding/showing viewers 521, 523–524

OpenScript, *continued*
importing books 488
importing delimited-field text files 478
importing fixed-field text files 476
importing graphics 231
importing pages 107, 490
importing pages with record fields 223
inserting inline graphics 215
limits to exporting scripts 50
list of menu-related commands 563–564
modifying menu bar resources 562–583
naming stages 345
object hierarchy 501–502
and objects 49
opening viewers 521
optimizing scripts 611, 618–623
playing animations 326–327
positioning viewers 515–520
previewing before printing 686
printing book pages 674–675
printing reports 679–680
and Property Browser 75, 188
removing clips 351
removing hotwords 269
removing media search paths 359
removing menu bars from viewers 562
removing menus and menu items 574
removing pages 112
removing resources 468
removing stored images 629
renaming menus and menu items 570–571
renumbering pages 111
replacing resources 464
restoring menu bars to original state 574

OpenScript, *continued*
restricting book navigation 506
saving page changes in viewers 526
setting background properties 114
setting button properties 253
setting chromakey property of paint objects 238
setting combo box properties 284
setting hotword properties 266–267
setting page size 117
setting printing conditions 674, 690–694
setting viewer properties 510, 514
sorting pages by record field 225–226
storing bitmap images 626
and viewers in applications 520–546
writing code for ActiveX controls 591–596
writing code for Automation objects 591–596
openWindow message 502
operators 384
order-text object 301
organizing application files for distribution 37–38, 353–356, 636–642
orientation, page 115

P

Package directory 649
Page Browser
illustrated 108
modifying page information 109
opening 108
overview 77, 107–108
uses for 107–108
viewing page information 109
page layout 685

pageNumber property 111
page orientation 115
pages, book
  adding 103–105
  adding hyperlinks to 148–150
  adjusting layout before printing 685
  backgrounds 44, 45
  displaying 94
  displaying in viewers 524–525
  displaying more than one 46
  displaying right-click menus 83
  illustrated 93
  importing 105–107, 223, 488–490
  moving objects between layers 166–172
  navigating 119–123
  optimizing 612–613
  overview 44
  previewing before printing 683–686
  printing 672–675
  removing 112
  renaming 109
  renumbering 109, 110–111
  reordering 109, 110–111
  reorganizing 107–112
  restricting navigation 506
  role of backgrounds 92–93, 103–105
  separate viewer windows as 46
  setting printing conditions in OpenScript 690
  skipping 121
  sorting by record field 223–226
  specifying size 115–117
  status bar indicator 63
  storing offscreen images 625–629
  transition effects 121–122
  viewing information 109
page size
  overview 115
  scaling windows to 117
  setting in Properties for Book dialog box 116
  setting using OpenScript 117
paint objects
  adding to books 229–233
  compared with picture objects 228
  converting picture objects to 236–237
  creating transparent areas 261–262
  cropping 233–234
  defined 228
  and DHTML 660
  importing 231
  transparent areas 237–238
palette flash 239, 457
Palette Optimizer 239, 240
palette property 458
palettes
  book palettes 239, 240, 241, 242, 457–458
  color tray 58, 66
  creating resource type 449
  defined 59
  floating tool, creating 541–543
  getting resource information 463
  hiding/showing 66–67
  illustrated 58, 66
  line ends palette 65
  line palette 65
  list of 65
  moving 67
  Palette Optimizer 239, 240
  pattern palette 65
  polygon palette 65



palettes, *continued*
tool palette 65–67, 145
as type of viewer 522–523
using 65–66
See also color palettes
Paragraph tab, Properties dialog box 182
parameters 385, 392, 396
parent objects
books as 502
compared with parent windows 502
defined 501
for viewers 502
parent windows
changing for viewers 533
compared with parent objects 502
parentHandle property 534
parentWindow property 533
passwords, controlling book access 118–119
passwordTypes function 119
pasting objects 151, 152
path animations 318, 320–321
paths, clip 353–356
pattern palette 65
pattern property 114
patterns
applying to backgrounds 113–114
filling objects 165
persistent properties 508–509
picture objects
adding to books 229–233
compared with paint objects 228
converting to paint objects 236–237
defined 228
importing 231
pie tool 145
placeholders 134
planning projects 37–41
playAnimation message 326
players. *See* media players
playStep message 326
plug-ins. *See* ToolBook II Neuron
polygon palette 65
polygons, drawing 146
polygon tool 145
pop-up menus 553, 582–583
pop-up windows
activating 528–529
compared with child windows 500
as viewers 497–498
positioning
objects on page 153–157, 613–614
viewers 514, 515–520
PowerPoint placeholder 134
previewing
overview 683
pages 683–686
reports 683–686
printall command 674
print command 674
printerArrangement property 675
printerConditions property 674
printerPageBitmap property 675
printers, list of properties 679
printing
applications 672–675
book pages 672–675
reports 675–680
scripts 694
setting conditions for 690–694
Print Report dialog box 677

programs. *See* online learning applications; Windows programs
progress bars, creating 543–546
project planning 37–41
properties
  adding user properties 190
  overview 47–48, 180
  persistent compared with nonpersistent 508
  and Property Browser 75, 188–190
  resources as 454–460
  setting for ActiveX controls 592–593, 597–598
  setting for Automation objects 592–593
  setting for books 118–119
  setting for buttons 249, 251–254
  setting for combo boxes 283–284
  setting for fields 197–201
  setting for hotwords 266–267
  setting for list boxes 276–279
  setting for objects 177, 183
  setting for OLE objects 607–608
  setting for question objects 294–296
  setting for stages 345–347
  setting for viewers 508–510
  setting using Properties dialog box 183
  setting using right-click menu 185–186
  See also extended properties
Properties dialog box
  as Author-level tool 59, 70–71
  compared with Property Browser 75, 188
  displaying 71, 181
  illustrated 70, 180
Properties dialog box, *continued*
  overview 70–71, 181
  tabbed pages 182, 183
  toolbar buttons 184–185
Property Browser 188–190
property pages 182, 183
Publish to Web 644, 662
pushbutton tool 145, 250

Q

question objects
  adding to pages 294
  arrange-objects object 297
  assigning weights to answers 303–304
  and DHTML 660
  dragging and dropping 129
  drag-objects object 298
  drop-targets object 299
  examples 297–301
  fill-in-the-blank object 300
  match-items object 300–301
  multiple-choice object 301
  order-text object 301
  overview 136, 292–294
  scoring questions 302–304
  selecting from Catalog 294
  setting extended properties 295–296
  setting properties 294–296
  specifying feedback 304–310
  viewing in Catalog 293
Quick Specialist 102
QuickTime files 335
quitting Instructor 25

R

radio button tools 145, 250
Reader level
  changes in interface 84–85
  compared with Author level 42, 84–85
  data entry 222
  editable compared with activated fields 200–201
  illustrated 57
  and menu bars 560
  navigating books 119, 120
  overview 42, 43
  playing animations 322–327
  right-click menus 83–84
  switching to Author level 85
readerStatusBar property 64
read-only access to Instructor books 652
RealMedia files 335
RealPlayer 335
record fields
  adding text 201–204
  applying color to text 211–212
  compared with text fields 192
  copying and pasting text 204, 206–208
  creating by dragging from Catalog 195
  creating using OpenScript 196
  creating using tool palette 196
  deleting text 208
  editable compared with activated 200–201
  exporting data to text files 479–484
  finding and replacing text 218–219
  formatting 209–214
  importing pages with 223
record fields, *continued*
  importing text into 203, 471, 473–478
  locking 200
  optimizing 617
  overview 192–194, 220
  setting properties 197–201
  sorting pages by 223–226
  tab order 169–172
  uses for 220–222
record field tool 145
rectangle tool 145
release notes 29
remove menu command 563, 574
remove menuItem command 563, 574
remove resource command 351, 468
removing
  ActiveX controls 590
  backgrounds 112
  clips 350, 351
  menus and menu items 574
  objects from Catalog 139
  objects from pages 153
  OLE objects 608
  pages 112
  resources 466–468, 639
renaming
  books 640–641
  menus and menu items 570–571
  pages 109
renumbering pages
  in Page Browser 109
  in Properties for Page dialog box 110
  using OpenScript 111

reordering pages
  by cutting and pasting 111
  in Page Browser 109, 110–111
  in Properties for Page dialog box 110
Replace dialog box 219
replacing resources 463–464
reports
  adjusting layout before printing 685
  columns compared with groups 675–676
  formatting text 681–683
  previewing before printing 683–686
  printing 675–680
  setting printing conditions in OpenScript 679, 691
reset page button 133
resetting books 641–642
reset trigger button 132
reshape handles 158–159
reshaping objects 158–159
resizing
  objects 157–158
  viewers 515
Resource Manager, defined 446
Resource Manager dialog box 450
resources
  adding to books 452–454
  assigning as object properties 454–460
  backdrops as 114
  copying 451, 453
  creating 449–451, 453
  duplicating 451
  exporting 465
  getting information 461–463
  graphics as 232, 447, 466
resources, *continued*
  importing 453, 454
  modifying 449–451
  overview 53–54, 446–448
  replacing 463–464
  storing in Instructor 446
  unused, removing 466–468, 639
  uses for 446
  See also color palettes; cursors; icons; menus; scripts
restore menuBar command 564, 574
rgbFill property 114
rgbStroke property 114
Rich Text Format (RTF) 204, 470
richText property 207, 622, 682
right-click menus
  defined 59
  displaying at Author level 82–83
  displaying at Reader level 83–84
  identifying graphic types 228
  illustrated 58, 82
  setting object properties 185–186
  toolbars for 82, 186
  using 81
rotating objects 154–155
rounded rectangle tool 145
RTF files 204, 470
rulers
  changing unit of measurement 80
  defined 59
  hiding/showing 80
  illustrated 58, 79
  using 79
run-time version of ToolBook II 651



S

saveOnClose property 526
saving
  backup copies 641
  books as templates 97–98
  books with new name 640–641
  page changes in viewers 526
Score field 133
score page button 133
score quiz button 133
scores, reporting 133, 302, 667
screens. *See* displays
script editor 257
  See also OpenScript
scripting buttons 257
scripts, OpenScript
  defined 49
  limits to exporting 50
  optimizing 611, 618–623
  placing handlers 623–624
  printing 694
  status bar indicator 64
searching for Catalog objects 130
select command 112
selecting
  groups 173
  objects 147–148
  text in fields 206
selection indicator 145
send command 530
send paste statement 208
separator bars, adding to menus 572
Setup directory 649, 651
setup programs, creating 651
shared action sequences 366, 388–396, 402, 426–440
shared scripts
  creating for navigation functions 623
  creating resource type 449
Shockwave Player 335
show command 64, 67, 521
showing. *See* hiding/showing
show menuBar command 564, 573
shown message 502
single-select list box tool 145
sized message 502
size property 117
sizeToPage message 117
sizing
  objects 157–158
  viewers 515
sizing handles 157, 158
skipNavigation property 121
Sort dialog box 224–225
sorting pages by record fields 223–226
sound files. *See* clips
special characters, adding to text 216–217
spelling, checking 638
splash screens, creating 547–548
spreading selected objects 154
stages
  changing appearance 345–347
  creating 344–345
  naming 344, 345
  setting properties 345–347
stage tool 145
starting Instructor 24–25
Startup dialog box, bypassing 99
startup screens, creating 547–548

status bars
  defined 59
  hiding/showing 64
  illustrated 58, 64
  indicators 64
  navigating pages using 63, 86–87
  for viewers 64, 514
stopAnimation message 327
stored offscreen images 625–629
storeImage property 626, 629
storing graphics files 233
streaming media 335
strokeColor property 114, 212, 235
submenus 552, 566
switching between backgrounds and book pages 92, 103
sysCursor property 461
sysFontFace property 683
sysFontSize property 683
sysFontStyle property 683
sysHistory property 525
sysHistoryRecord property 615
sysLockScreen property 527, 615
sysOpenWindows property 522
sysSuspendMessages property 615
system cursors
  changing 461
  removing references to resources 467
system requirements 22, 23
system variables 620–621

T

TAB key 169
tab order, planning 169–172
tabSpacing property 682
target windows
  changing temporarily 529–530
  changing using *send* command 530
  closing 532
  compared with focus windows 527
  overview 527–528
  referring to objects in 531–532
technical support 30
Template directory 97
templates
  basing new books on 99–100
  creating for Instructor books 97–98
  designating default 96–97
  illustrated 95
  optimized for Internet 96
  overview 94–96
  saving books as 97–98
  starting new books from 99–100
  for viewers 503, 504
testing applications
  CD-ROM installations 654
  how to do it 653
  importance of 124
text
  adding to combo boxes 284–285
  adding to fields 201–204
  adding to list boxes 279–281
  applying color 211–212
  copying and pasting 204, 206–208
  deleting from fields 208
  exporting 471, 472–473, 479–484
  file formats for importing and exporting 470

text, *continued*
finding and replacing in fields 218–219
formatting character styles 210–211
formatting paragraph styles 213–214
formatting reports 681–683
importing into record fields 471, 473–478
importing into text fields 203, 471–472
inserting inline graphics 214–215
keyboard shortcuts for navigating 205
optimizing in fields 617
optimizing in scripts 621–622
selecting in fields 206
wrapping within fields 199–200
See also text fields
text expressions 383
text fields
adding text 201–204
applying color to text 211–212
compared with record fields 192
copying and pasting text 204, 206–208
creating by dragging from Catalog 195
creating using OpenScript 196
creating using tool palette 196
deleting text 208
and DHTML 660
editable compared with activated 200–201
exporting text from 471, 472–473
finding and replacing text 218–219
fixed compared with delimited 471
formatting 209–214
importing text into 203, 471–472
locking 200
optimizing 617
overview 192–194
text fields, *continued*
setting properties 197–201
tab order 169–172
text files
exporting record field data to 479–484
importing into record fields 471, 473–478
importing into text fields 471–472
textline property 280
text property 202, 207, 280
TIFF files 233
tiling child windows 498, 499
time markers, media 337–343
toolbars
defined 59
hiding/showing 63
illustrated 58
list of buttons 61–62
Properties dialog box 184–185
right-click menu 82, 186
using 60
ToolBook II
and learning management systems 667–669
run-time version 651
ToolBook II Animation Editor
creating cel animations 321–322
creating path animations 320–321
modifying animations 327–329
opening 319
overview 318–319
testing cel animations 322
testing path animations 321
ToolBook II Assistant 34
ToolBook II Book Specialists. *See* Book Specialists

ToolBook II Coach 27–28, 59, 69
ToolBook II Instructor benchmarking application performance 630–631
defined 34
distribution file list 29
documentation 26–28
exiting 25
installing 24
online resources 27–30
overview 20, 35–37
package contents 21
read-only access to books 652
release notes 29
starting 24–25
system requirements 22, 23
technical support 30
visual interface 56–86
See also online learning applications
ToolBook II Instructor Catalog. *See* Catalog
ToolBook II Instructor Startup dialog box 25
ToolBook II Neuron
as book distribution option 642, 643, 658, 661
Impulse utility 661
overview 34, 661
ToolBook II Web Specialist. *See* Web Specialist
tool palette
creating buttons using 250
creating combo boxes using 282
creating fields using 196
creating list boxes using 275
creating objects using 144–147
creating viewer to display 541–543
tool palette, *continued*
floating 541–543
illustrated 145
overview 65–67
tools, interface. *See* interface, Instructor
transition effects
and media clips 347
specifying 120–121
transparent areas 114, 237–238, 261–262
transparent objects 164
triggers. *See* action buttons
troubleshooting
action sequences 441
ActiveX controls 601
Automation objects 602

U

uncheck menuItem command 564, 571
ungrouping objects 178
uninstalling applications 654
units of measurement 80, 81, 116
Universal Media Player
adding to applications from Catalog 336
choosing media files 336–337
creating actions for time markers 340–343
list of supported media types 335
overview 333, 335
and time markers 337–340
See also media players
unused resources, removing 466–468, 639
URLs, creating hyperlinks to 149, 150
useChromakey message 238
User event 375–376
user properties 190

V

variables 385, 386–388, 392, 414–421, 620–621
vector graphics, as picture objects 228, 233
video players 135, 332, 334, 335
viewers
  activating 528–529
  assigning icons to 511
  assigning image buffers to 512–513
  assigning menu bars to 512, 560–561
  behavior of 497–498
  changing Minimize icon 460
  child windows 497–498
  client windows 500, 501
  closing 521, 522
  copying and pasting 507
  creating using templates 503, 504
  custom 503, 504
  default pages 495
  and DHTML 660
  displaying dialog boxes 535–541
  displaying floating tool palettes 541–543
  displaying pages 524–525
  displaying progress bars 543–546
  displaying splash screens 547–548
  elements of 500–501
  focus windows 527, 528–529
  frame windows 500, 501, 534
  hiding/showing 521
  illustrated 495
  Internet compatibility 494
  locking screen 526–527
  message handling 502
  modal compared with nonmodal 523
viewers, *continued*
  opening 521
  overview 46, 494–501
  palettes as type of 522–523
  parent objects 502
  parent windows 502, 533, 534
  persistent compared with nonpersistent properties 508
  pop-up windows 497–498
  positioning 514, 515–520
  resizing 515
  right-click menus 83
  role in object hierarchy 501–502
  saving page changes 526
  setting caption style 511
  setting mat color 512
  setting properties 508–510
  setting size 515
  showing 521
  status bars for 64
  target windows 527–528, 529–532
  types of 496
  uses for 494
  using color palettes with 522–523
  using handles 534
  using in applications 520–546
Visual Basic compared with OpenScript 591–592
visual effects. *See* animation; animations; hide-and-show effects
visual interface. *See* interface, Instructor
VRML placeholder 134



W

Web pages
  click2learn.com 28, 29–30
  creating hyperlinks to 149, 150
  distributing Instructor applications as 40, 643, 644, 656–666
Web site, click2learn.com 28, 29–30
Web Specialist 642, 643, 644, 656, 662–663
weighting answers 303–304
windowHandle property 534
windows
  child windows 497–498, 499, 500
  client windows 500, 501
  focus windows 527, 528–529
  frame windows 500, 501, 534
  parent windows 502, 533, 534
  pop-up windows 497–498, 500, 528–529
  scaling to page size 117
  target windows 527–528, 529–532
  ToolBook II Catalog 126
  See also viewers
Windows functions. *See* dynamic-link libraries (DLLs)
Windows Media Player 335
Windows programs
  including OLE objects in Instructor applications 51, 603–604
  Instructor application support 50
Windows system colors 163–164
WMF files 228
wrapping text within fields 199–200

CLICK2LEARN.COM, INC.
TOOLBOOK II SOFTWARE LICENSE AGREEMENT

THIS IS A LEGAL AGREEMENT BETWEEN YOU AND CLICK2LEARN.COM, INC.

("CLICK2LEARN.COM") WHICH GOVERNS YOUR USE OF THIS SOFTWARE. CLICK2LEARN.COM IS WILLING TO LICENSE THE SOFTWARE TO YOU ONLY ON THE CONDITION THAT YOU AGREE TO ALL OF THE TERMS OF THIS AGREEMENT. CAREFULLY READ THIS AGREEMENT BEFORE INSTALLING OR USING THE SOFTWARE. BY INSTALLING OR USING THE SOFTWARE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE SOFTWARE TO THE PLACE YOU OBTAINED IT FOR A FULL REFUND.

LICENSE. Under this license you may install and use the enclosed click2learn.com software program ("Software") on only a single computer at any one time. You may also make a single copy of the Software for archival purposes, or copy the Software to a single hard disk provided you retain the original for archival purposes and provided that copyright and other restricted rights notices are also reproduced. The Software is "in use" when it is loaded into RAM, or installed on a hard disk or other storage device (other than a network file server). In addition, you may create or modify ToolBook II Assistant objects using ToolBook II Instructor; provided that you may not create or modify any objects such that scripts may be modified or created using ToolBook II Assistant.

You may not (i) install the Software on a network file server unless you have a serialized Additional User Software License Agreement for each user accessing the Software through the network; (ii) copy the written materials accompanying the Software; (iii) use, copy, modify, translate, or merge the Software except as provided in this agreement; (iv) reverse-engineer, disassemble, or decompile the Software or otherwise attempt to determine its underlying source code or permit any such actions except and to the extent expressly permitted by the application of local copyright law whose exclusion is precluded; or (v) sublicense (except to the extent specifically permitted herein), rent or lease all or any portion of the Software, but you may transfer the Software and manuals to a single recipient on a permanent basis provided you retain no copies, the transfer includes all updates and prior versions of the Software licensed to you and the recipient agrees to the terms of this agreement.

COPYRIGHT. The Software is owned by click2learn.com or its suppliers and is protected by United States copyright laws and international treaty provisions. You must treat the Software like any other copyrighted material.



REDISTRIBUTION. The Software includes one or more of the following classifications of Redistributable Files, as defined in the file named DISTRIB.WRI: click2learn.com Runtime Files, Additional Runtime Files and Clip Files. To the extent provided below in this section and subject to any additional restrictions for particular Redistributable Files as described in DISTRIB.WRI, this license gives you (but not licensees of your Application) a limited right to copy and distribute Runtime Files and to modify, copy and distribute Clip Files, on condition that:

(i) you copy or distribute the Additional Runtime Files and Clip Files only in conjunction with, linked to, and as a part of your applications created with the Software or operating in an click2learn.com ToolBook II environment ("Application"); except that you may not use or permit the use of any Clip Files in a library of clip files—you may distribute the click2learn.com Runtime Files separately from your Application;

(ii) you prominently display the following notice on the splash screen, initial screen, exit screen or in the about box of your Application, excluding any Application consisting solely of DHTML output, but including those displayed through Neuron: **"Created with ToolBook II from click2learn.com";**

(iii) you include the following notice on the copyright screen of your Application and where ever your copyright notice appears in the documentation and packaging for your Application: "Portions © 1990-2000 click2learn.com, inc. All Rights Reserved.";

(iv) you do not remove any copyright notices from the Redistributable Files;

(v) if you distribute your Application outside your organization, you do so only pursuant to a license agreement between you and the users of your Application that prohibits any use prohibited by this license (including subsequent redistribution of Redistributable Files by such licensees of your Application), and that contains provisions that disclaim your liability for consequential and incidental damages and that provides that such provisions expressly inure to the benefit of your suppliers;

(vi) you indemnify and hold harmless click2learn.com and its suppliers from and against any damages or liabilities, including attorneys' or other legal fees, that arise or result from claims or lawsuits regarding the use or distribution outside your organization of your Application;

(vii) if you distribute your Application outside your organization, you include click2learn.com's restricted government rights notices prominently on your Application and license agreement; AND

(viii) you comply fully with the terms and conditions of this license and all applicable laws and regulations.



Unless you receive the prior permission of click2learn.com, this license does not permit you to create or distribute an Application that (a) can be used to create new applications designed to operate in a click2learn.com ToolBook II environment, or (b) can be used to create a DHTML application, or (c) provides the ability to create scripts or to copy predefined scripts to new objects, or (d) provides the ability to define or redefine hyperlinks. You may contact click2learn.com at the address below to request permission to create and distribute such an Application.

LIMITED WARRANTY. Click2learn.com warrants that, for 90 days after date of delivery of the Software to you, the disks or CD-ROM (if the Software was delivered on tangible media) will be free from defects in materials and workmanship. Your exclusive remedy for breach of warranty is to return the complete product, with proof of purchase, to click2learn.com (if purchased direct) or to your reseller and it will be replaced for free. You are responsible for the selection and use of the Software, and, unless you have separately purchased installation support from click2learn.com, for the installation and configuration of the Software. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLICK2LEARN.COM AND ITS SUPPLIERS EXPRESSLY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION WARRANTIES AND CONDITIONS OF MERCHANT-ABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT, AS WELL AS ANY WARRANTIES THAT THE OPERATION OF THE PRODUCTS WILL BE INTERRUPTION OR ERROR FREE. EXCEPT AS EXPRESSLY SET FORTH HEREIN, THE SOFTWARE IS PROVIDED "AS-IS." ALL WARRANTIES AND CONDITIONS SHALL TERMINATE 90 DAYS FROM DATE OF DELIVERY OF THE SOFTWARE TO THE ORIGINAL LICENSEE. Some jurisdictions do not allow limitations on how long an implied warranty or condition lasts, so the above limitation may not apply to you. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY HAVE OTHER RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION. IMPORTANT NOTE: Nothing in this agreement is intended or shall be construed as excluding or modifying any statutory rights, warranties or conditions which are applicable to this agreement or the Software, and which by virtue of any national or state Fair Trading, Trade Practices or other consumer legislation may not be modified or excluded. If permitted by such legislation, however, click2learn.com's liability for any breach of any such warranty or condition shall be and is hereby limited to either: (i) the supply of such Software again or; (ii) the correction of any defect in such Software as click2learn.com at its sole discretion may determine to be necessary to correct the defect.

EXCLUSIVE REMEDY. EXCEPT AS OTHERWISE PROVIDED HEREIN, YOUR EXCLUSIVE REMEDY AND CLICK2LEARN.COM'S AND ITS SUPPLIERS' ENTIRE LIABILITY ARISING FROM OR IN CONNECTION WITH THE SOFTWARE, DOCUMENTATION AND/OR THIS AGREEMENT SHALL BE THE REPLACEMENT OF THE SOFTWARE.



LIMITATIONS OF LIABILITY. IN NO EVENT WILL CLICK2LEARN.COM OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR EXEMPLARY DAMAGES, ARISING OUT OF OR IN CONNECTION WITH YOUR USE OR INABILITY TO USE THE SOFTWARE OR DOCUMENTATION, THE BREACH OF ANY EXPRESS OR IMPLIED WARRANTY OR CONDITION, OR OTHERWISE IN CONNECTION WITH THE SOFTWARE, THE DOCUMENTATION , AND/OR THIS AGREEMENT EVEN IF CLICK2LEARN.COM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Some jurisdictions do not allow limitation or exclusion of incidental or consequential damages, so the above limitation or exclusion may not apply to you to the extent that liability is by law incapable of exclusion or restriction. IN NO EVENT SHALL CLICK2LEARN.COM'S OR ITS SUPPLIERS' TOTAL LIABILITY FOR ANY DAMAGES, DIRECT OR INDIRECT, IN CONNECTION WITH THE SOFTWARE, THE DOCUMENTATION, AND/OR THIS LICENSE EXCEED THE LICENSE FEES PAID FOR YOUR RIGHT TO USE THIS COPY OF THE SOFTWARE WHETHER SUCH LIABILITY ARISES FROM ANY CLAIM BASED UPON CONTRACT, WARRANTY, TORT OR OTHERWISE. Provisions of this agreement such as the warranty limitations, exclusive remedies and limitations of liability are unrelated, independent allocations of risks between you and click2learn.com. Unenforceability of any such allocations shall not affect the enforceability of other such allocations. If any part of this agreement is held to be unenforceable, it shall not affect any other part. If any part of this agreement is held to be unenforceable as written, it shall be enforced to the maximum extent allowed by applicable law. click2learn.com's pricing reflects the allocations of risk contained in this agreement.

GENERAL. You agree not to directly or indirectly export or re-export the Software or the direct product thereof except as authorized by the laws and regulations of the United States and any other applicable jurisdiction. This license and your right to use the Software terminate automatically if you violate any part of this agreement. In the event of termination, you must immediately destroy all copies of the Software or return them to click2learn.com. This agreement constitutes the entire agreement between you and click2learn.com and supersedes any prior agreement concerning the contents of this package. It shall not be modified except by written agreement dated subsequent to the date of this agreement signed by an authorized click2learn.com representative. Click2learn.com is not bound by any provision of any purchase order, confirmation, correspondence or otherwise, unless click2learn.com specifically agrees to the provision in writing. This agreement is governed by the internal laws of the State of Washington, U.S.A., and specifically excludes the United Nations Convention on Contracts for the International Sale of Goods.



U.S. GOVERNMENT RESTRICTED RIGHTS.

The Software and written materials accompanying the Software are provided with RESTRICTED RIGHTS. If you are an agency of the United States Government, the following clause applies to this license. The Software is commercial computer software developed at private expense. Software provided to the United States Government pursuant to solicitations issued on or after December 1, 1995 is provided with the commercial license rights and restrictions described elsewhere herein. All Software provided to the United States Government pursuant to solicitations issued prior to December 1, 1995 is provided with “Restricted Rights” as provided for in the Commercial Computer Software-Restricted Rights at FAR, 48 CFR 52.227-14 (JUNE 1987) or The Rights in Technical Data and Computer Software clause at DFAR, 48 CFR 252.227-7013 (OCT 1988), as applicable. Use, duplication or disclosure by the United States Government is subject to restrictions as set forth therein. Manufacturer is click2learn.com, inc., 110 - 110th Avenue N.E., Bellevue, Washington, U.S.A. 98004.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

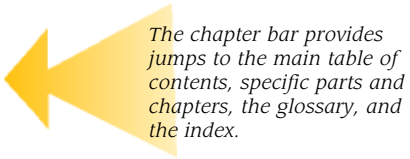
Index

How to use this online book

This online book is designed to help you access information quickly. Use the features described here to jump to the chapter, topic, or resource you need.

Along the left side of every page is the chapter bar, where you can click a chapter you want to look at and jump immediately to the first page of that chapter. At the lower right of each page are navigation buttons that move you to the next or previous page.

On the first page of each chapter is a list of the chapter's contents. Click a topic or its page number to jump directly to that topic.



The chapter bar provides jumps to the main table of contents, specific parts and chapters, the glossary, and the index.

CONTENTS

Navigating from the main table of contents **G.2**

Navigating from the index **G.3**

Using Adobe Acrobat Reader **G.4**



The chapter table of contents provides jumps to main topics within a chapter.

Navigation buttons take you to the next or previous page.



How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index



The main table of contents provides jumps to parts and chapters, main topics, the glossary, and the index.

Navigating from the main table of contents

In addition to the chapter bar and the chapter table of contents, you can navigate using the main table of contents, which lists all the chapters in the book and the main topics of each chapter. To quickly jump to the main table of contents, click Contents at the top of the chapter bar.

From the main table of contents, click a chapter title or chapter topic to jump immediately to that chapter or topic.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Navigating from the index

The index lists key words and phrases in this guide. Each item is linked to the page where it appears. From the index, click a word or phrase to jump to the location of the word or phrase.

Index

A

Adobe Acrobat Reader G.4

B

Button, Find G.6
Buttons, navigation G.4

C

Chapter bar G.1

F

Find button G.6
Find dialog box G.6
Forms G.6

N

Navigating G.2, G.3
Navigation buttons G.1

P

Print G.6

R

Reader Online Guide G.4, G.6

S

Search G.6
Setup G.4

T

Text selection G.5
Toolbar, Acrobat G.4
Tools
  page sizing G.6
  Text selection G.5
  Zoom in G.5
  Zoom out G.5

W

Web site G.6

Z

Zoom-in tool G.5
Zoom-out tool G.5

The index provides jumps from key words and phrases.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

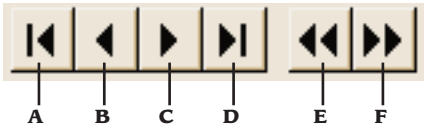
Using Adobe Acrobat Reader

Adobe® Acrobat® Reader is a software program that allows you to view this online guide. Acrobat Reader is installed on your computer during Setup. It comes with a Reader Online Guide, available from the Help menu, which explains how to use the program. The Acrobat toolbar provides some helpful features, described in the next section.

The Acrobat toolbar

In addition to the navigation features in this guide, you will find useful navigation tools on the Acrobat toolbar.

Navigation buttons



A First page
B Previous page
C Next page
D Last page
E Previous view
F Next view

The First, Previous, Next, and Last buttons allow you to move quickly within the online book. Additionally, Acrobat allows you to move to the previous page visited and back again using the Previous view and Next view buttons. This type of navigation is convenient when jumping between chapters or nonsequentially within a chapter.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

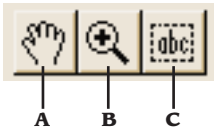
PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Zoom and text tools



A *Hand tool*
B *Zoom-in tool*
C *Text selection tool*

Acrobat also comes with tools you can use to zoom in and out of a page, and to move about a page. Use the Hand tool to move around the page by clicking and dragging. The Zoom-in tool magnifies the page with each click. Select the Zoom-in tool and hold down the CTRL key to show the Zoom-out tool.

Another useful tool is the Text selection tool. Use this feature to select a block of text. Then, copy the text to another program.

Here's a tip for those of you with tired eyes.

Grab the Zoom-in tool and select this paragraph (as shown). This will magnify the page to the specified area. Use the scroll bars or the Hand tool to move around as you read.

How to use this online book

Contents

PART I
Welcome to ToolBook II Instructor

PART II
Building your application

PART III
Adding interactive features to your application

PART IV
Adding behavior with action sequences

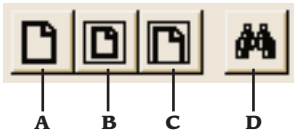
PART V
Using advanced features

PART VI
Distributing and printing your application

Glossary

Index

Page-sizing tools



A Zoom 100%
B Fit page
C Fit width
D Find

Using the page-sizing tools, you can quickly change the size of the page you are viewing. This feature comes in handy when you want to restore the view of the whole page after using the Zoom-in tool. This guide is designed to be viewed at the Fit page size. At this size, you can see all of the navigation tools available to you.

The Find button opens the Find dialog box, where you can enter a word or phrase to search for.

Additional Acrobat features

Using features in Adobe Acrobat, you can do many other things, such as print the document, fill in forms, jump to a Web site, and open other files. For more information on these and other features, see the Reader Online Guide, available from the Help menu.