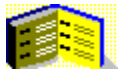


Sample Applications Guide Contents



[Sample applications \(by type\)](#)

[Sample applications \(by file name\)](#)



Sample applications (by type)

The topics below briefly describe each sample application and its contents.

To find sample applications organized by file name, go to [Sample applications \(by file name\)](#).

Animation

[CBT.TBK](#)

[FEATURES.TBK](#)

Computer-based training (CBT) templates

[CBT.TBK](#)

[CBT1.TBK](#)

[CBT2.TBK](#)

[CBT3.TBK](#)

[CBT4.TBK](#)

[CBT_DEMO.TBK](#)

Databases

[CONTACT.TBK](#)

[DBEXCHNG.TBK](#)

[WHOWHERE.TBK](#)

Demos

[CBT1.TBK](#)

[CBT2.TBK](#)

[CBT3.TBK](#)

[CBT4.TBK](#)

[CBT_DEMO.TBK](#)

[CBT.TBK](#)

[KIOSK.TBK](#)

[PRESENT.TBK](#)

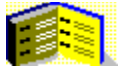
Help and online documentation

[TMPLHELP.TBK](#)

Hypertext and content-rich applications

[HANDBOOK.TBK](#)

MAPI[LIBRARY.TBK](#)[MAPI.TBK](#)**Object and text editors**[AUTOEDIT.EXE](#)[DIALOG.TBK](#)**Object and resource browsers**[BROWSER.EXE](#)[LIBRARY.TBK](#)[SCRAPBK.TBK](#)**Windows API**[LIBRARY.TBK](#)[SCRAPBK.TBK](#)



Sample applications (by file name)

The topics below briefly describe each sample application and its contents.

To find sample applications organized by type, go to [Sample applications \(by type\)](#).

[AUTOEDIT.EXE](#)

[BROWSER.EXE](#)

[CBT.TBK](#)

[CBT_DEMO.TBK](#)

[CBT1.TBK](#)

[CBT2.TBK](#)

[CBT3.TBK](#)

[CBT4.TBK](#)

[CONTACT.TBK](#)

[DBEXCHNG.TBK](#)

[DIALOG.TBK](#)

[FEATURES.TBK](#)

[HANDBOOK.TBK](#)

[KIOSK.TBK](#)

[LIBRARY.TBK](#)

[MAPI.TBK](#)

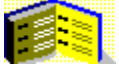
[PRESENT.TBK](#)

[SCRAPBK.TBK](#)

[TAQUIN.TBK](#)

[TMLPHELP.TBK](#)

[WHOWHERE.TBK](#)



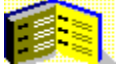
AUTOEDIT.EXE

The AutoScript editor is a tool that provides an interface for using AutoScript files. Using this editor, you can modify, add, remove, and search for scripts in an AutoScript file (.ATS). The tool bar at the top of the page is an `alwaysReader` viewer that provides shortcuts to many functions in the editor.

Features of this sample application:

- ♦ To create a new AutoScript file, click "No" as the response to the initial question for opening the default ATS file.
- ♦ The button bar that allows you to choose whether to show the captions on the buttons.
- ♦ Popup help appears for most objects after the mouse pauses over the object. The `mouseEnter` message sets a time that the `idle` message keeps track of. If a certain period of time passes (approximately 0.5 seconds) before the `mouseLeave` message, the text of a field is set to the help text associated with the object, and the viewer is sized to show this text.
- ♦ A number of the dialog boxes use the same viewer, but the page the viewer is set to varies. The viewer acts as a frame for pages used as dialog boxes.
- ♦ The lock in the field is actually a button with a graphic and no caption. When you click the button, ToolBook toggles the graphic and the behavior of the field changes.
- ♦ Standard Windows Open and Save dialogs are used for the AutoScript files.





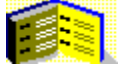
BROWSER.EXE

The Multimedia Browser utility allows you to play multimedia files directly from your hard drive. Its interface is similar to a File Open dialog box with fields that display directories, files, and drives on the system. It is possible to filter files by type, such as .EXE or .WAV. Fields at the bottom left of the window display type, length, and size information for the currently selected file.

Features of this sample application:

- ♦ When the script in the stage object on the left side of the page receives a `showMedia` message, it analyzes a media file and determines if it is visual. If the file is visual, it is shown in the stage, and the play and stop controls are activated.
- ♦ A single clip reference is used for this book. When a different media device is selected, the `mmSource` property of that clip changes.
- ♦ The Multimedia Browser makes extensive use of notify handlers. Most of the scripts are kept with the individual objects. This book contains very few scripts.



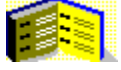


CBT.TBK

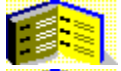
The CBT sample application is an example of a shell for a corporate CBT program. The book contains a menu page, four quiz pages, and a score page. The question properties can all be set through the Widgets tab of the Property editor. The questions are designed to provide automatic feedback and scoring.

Features of this sample application:

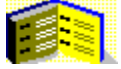
- ♦ The application sizes itself to run at full screen.
- ♦ Each page has a property that the buttons on the menu page check to indicate automatically which pages have been visited.

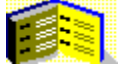


The questions all use a shared script resource and are designed to have properties set through the Widgets tab of the Property editor.



Question #4 uses an .AVI file as the input device for the multiple-choice question.

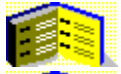




CBT_DEMO.TBK

The CBT Demo book is an example of using the template, CBT2.TBK (included with Multimedia ToolBook), and turning it into a complete application by adding content. The content for this application covers repetitive stress injuries (RSI).

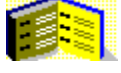
Features of this sample application:



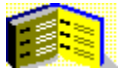
Popup help using hotwords with graphics is provided on page 6.

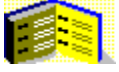


Hypernavigation using hotwords is demonstrated on page 9.



Stages showing AVI video can be found on pages 10 and 11.

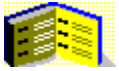




CBT1.TBK

This template is designed for menu-based applications which are divided into chapters and sections within each chapter.

Features of this sample application:

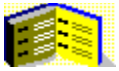


Switching to Author level displays the Template Outline viewer which is used to add or remove pages in the template. (If the Template Outline viewer is not displayed in Author view, choose Template Outline from the Contents menu.)

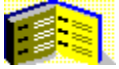
To change the title, select an existing chapter or topic and type a title in the Edit Title field.

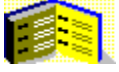
To add a chapter or topic, click New at the bottom of the viewer, select Chapter or Topic in the New dialog box, and then click OK. When you add a chapter or topic, ToolBook automatically creates and displays an untitled page, which you can then name. New chapters are placed at the end of the book; new topics are placed after the currently selected chapter or topic.

To remove a chapter or topic, select the title and click the Delete button at the bottom of the viewer.



Section names are displayed in the tool bar viewer that is displayed as a tiled viewer at the top of the page. This viewer's `alwaysReader` property is set to `true` so it can be used at Author level as well.





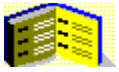
CBT2.TBK

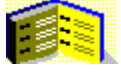
This template is used in CBT_DEMO.TBK. Use this template to create applications that cover a specific content, such as mountain climbing or product features. It uses a unique tab control that generates the illusion that the tabs hang below the window. This is accomplished by setting the background color as well as the mat color to black and maximizing the ToolBook window. The tabs themselves are part of a group on the background.

A book script is included that guarantees an application opens at full-screen. This script can be copied from below:

```
to handle author
  set state of mainWindow to normal
  set captionBar of mainWindow to normal
  forward
end
to handle reader
  forward
  set captionBar of mainWindow to none
  set state of mainWindow to maximized
end
```

Note The Author level menu has been altered in this book.

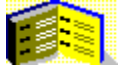




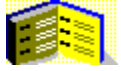
CBT3.TBK

This template is an example of a timeline. Each screen of the application is seen in a viewer in the center of the main window. When the user clicks one of the navigation buttons, ToolBook changes the `currentPage` property of the viewer using a `push` transition effect. The transition script uses both "`push right`" and "`push left`" to create the effect of one long page that is being scrolled left and right rather than several different pages. You can find the script for the transition effects in the background timeline.

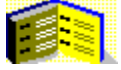
Features of this sample application:



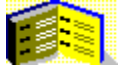
When the user clicks a character button, ToolBook displays a biographical sketch. Each button has a name and a corresponding page with a matching name. When the user clicks a button, ToolBook displays the viewer "`biography`" with its `defaultPage` property set to the button's corresponding page. The corresponding page displays a field with information on the button's character.

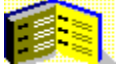


The background "`biographies`", which contains pages displayed by the "`biography`" viewer, contains a `windowSized` handler that dynamically resizes both the field and the background when the window is resized.



The main window's background contains tiled buttons with `normalGraphic` set to a bitmap resource.

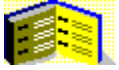




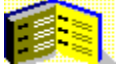
CBT4.TBK

This template features a navigation viewer with the `tile` property set to `left`. The user can resize this viewer can be dynamically by moving the separator bar (which is a button on the page) to the left or right. The script for resizing the viewer and all the objects within it is in button "`splitter`" of page "Navigation".

Features of this sample application:

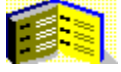


In addition to navigation buttons that move left and right through the book, there is a search button (indicated by binoculars). When the user clicks the search button, ToolBook displays a dialog box in which the user can enter text to search for.



The navigation viewer's `alwaysReader` property is set to `true`, but at Author level, it contains extra options for adding and removing topics in the template. You can add, delete, and modify existing categories and topics. When you add or delete topics, ToolBook automatically adds or removes pages in the template.



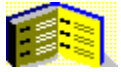


CONTACT.TBK

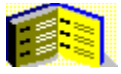
This is a fully functional interface to a dBASE III file. It includes all links to TB40DB3.DLL which is the dBASE III engine, and contains wrappers (handlers) for important DLL functions located in the book script. It uses a single table called CONTACT.DBF and two indexes, COMPANY.NDX and NAME.NDX. Simple controls allow you to modify, add, or delete records in the table.

Because the database is indexed based on name and company, you can search by either of these fields. The book uses both Edit and Browse modes. In Edit mode, data can be updated, added, or deleted. In Browse mode, the data can only be viewed.

Features of this sample application:

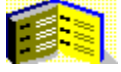


Help in the status bar is provided by using a `mouseEnter` handler and a user property `helpText`.

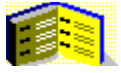


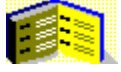
Text strings for Help and other interface elements are included in the book script to aid in

translation.



The Main window changes sizes when switching between Edit and Browse modes. There are two user properties of the Main window, `editView` and `browseView`, which store the size of the Main window for the two modes.

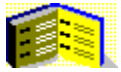




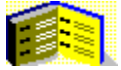
DBEXCHNG.TBK

DBASE Exchange is an import and export utility for dBASEIII and Paradox tables. There are two ways to import both Paradox and dBASEIII tables. You can import a table to a ToolBook front end, or to a ToolBook database that uses record fields for each field in the table and a different page for each record. If a ToolBook database is chosen, the database file itself is used only to build the database, but is no longer used after the ToolBook application is created.

Features of this sample application:

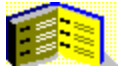


Tab controls allow users to navigate among the various options.

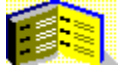


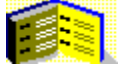
file.

The handler `linkPX` found in the book's script can be accessed in the PARADOX.ATS AutoScript



Database Exchange uses Windows Open and Save As dialog boxes. These dialog boxes are accessed via the DLL functions `openFileDialog()` and `saveAsDlg()` in TB40DLG.DLL.



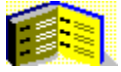


DIALOG.TBK

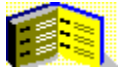
The Dialog editor is a utility that can create dialog boxes. You can create two types of dialog boxes with the Dialog editor: Windows dialog boxes or ToolBook viewers that function as dialog boxes. Windows-style dialog boxes are displayed using the `dialog()` function in the DLL TB40DLG.DLL. A viewer dialog box is a ToolBook viewer with a dialog box frame displayed as `modal`.

The Dialog editor is composed of a tool bar with basic functions such as opening and closing existing dialog boxes, duplicating objects, and previewing the dialog box; a tool palette with various objects that you can place in the dialog box; and an empty dialog box frame that represents a working model of the dialog box.

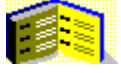
Features of this sample application:



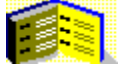
The Dialog editor uses Windows Open and Save As dialog boxes. These dialog boxes are accessed via the DLL functions `openFileDialog()` and `saveAsDlg()` in TB40DLG.DLL.



ToolBook writes scripts dynamically to a target application when you assign a dialog box to a book and, in the case of viewer dialog boxes, a viewer is dynamically created. ToolBook stores the information for the dialog boxes as user properties in the target application. Windows dialog boxes require the use of TB40DLG.DLL to be displayed.

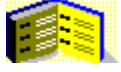


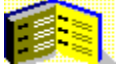
There is an option to save the dialog box as a .DIA file. This file contains the control codes that are passed to the `dialog()` function in TB40DLG.DLL.



To show the main window, choose Edit Editor from the Help menu. Right click on anything but the tool palette. Select the book DIALOG.TBK from the parent menu. Click the edit script button in the right-click menu. In the `enterApplication` handler comment out the line `hide mainWindow`. Select Update and Save Book (Ctrl-B) from the File menu. Close the Script editor, close the book, and re-open it.

Note This book is always at Author level.

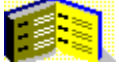




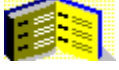
FEATURES.TBK

This sample application describes the features available in Multimedia ToolBook 4.0. Every page in this book has useful information and suggestions on how to use ToolBook. The book itself has many useful components. Each navigation button simulates a rubber button being pushed in, including the associated sound.

Features of this sample application:



Provides many examples of viewers, including: Progress Box, Dialog Box, Floating Palette, TV screen, and Popup explanation with graphics.



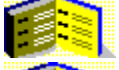
Shows a simple example of using drag-and-drop. Notice the script in the field and the drag-and-drop properties of the field and the "trash" object.



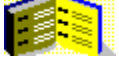
Provides scripts for manipulating arrays.

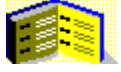


Includes an example of an analog clock based on notify handlers.



Includes Open File, Save As, Choose Color, and Choose Font dialog boxes.





HANDBOOK.TBK

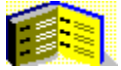
The employee handbook application is an example of flowing formatted text into a ToolBook application. The template used in this book has two record fields in the Main window. The first record field contains the title or heading, and the second contains information on the heading. A tiled viewer (to the right of the Main window) contains navigation buttons for moving backward and forward through the application, and an outline style widget that expands and contracts when items identified by a plus (+) or minus (-) sign are double-clicked.

To reuse this field, switch to Author level, open the Command Window (Shift+F3), and type:

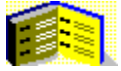
```
go to page "topics"
```

Then press Enter. This will take you to the page displayed in the navigation viewer. At this time, two copies of the navigation palette will appear in the Main window. The one to the right is the actual viewer with the `alwaysReader` property set to `true` (objects in that window cannot be selected). Select the field in the middle of the page on the left. This is a field called "List". Choose Copy from the Edit menu to copy the field to the Clipboard. You can paste this into another ToolBook application by opening the application and choosing Paste from the Edit menu.

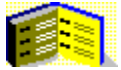
Features of this sample application:



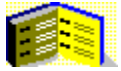
Uses a progress bar to indicate status when flowing the text.



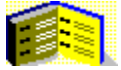
Includes an example of adding bookmarks. The script for adding bookmarks is located in `group id 5 of background "content"`.

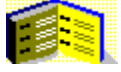


Provides status Help for all major components.



Includes useful `enterMenu` script, typical of how this message is handled, which handles the setup of certain menus by context.

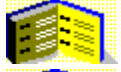




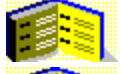
KIOSK.TBK

The kiosk sample application demonstrates a method of presenting information to a customer in a format that is pleasing and easy to use. Windows timers provide an automatic mode that, when enabled, plays background music and turns the page every 15 seconds.

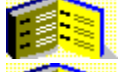
Features of this sample application:



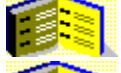
The application sizes itself to run full screen.



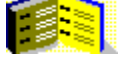
The menu page updates itself to reflect the current book.

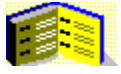


Windows timers are used to control timed events.



Background music plays when automatic mode is enabled.



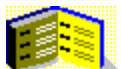


LIBRARY.TBK

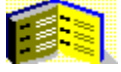
This book contains many different types of objects and examples. You can view this collection of objects and scripts by category. The complete list of widgets and scripts follows:

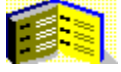
Add and Remove	Matching Quiz
Add number of days to date	Max Select Field
Alpha Keyboard	Money
Auto Buttons	Mouse browsing
Auto Page Name	Multiple Choice
Auto Page Number	Multiple OLE Drag and Drop
Auto-scroll Field	Object Effect
Auto-scroll Recordfield	Outline
Better Combobox	Outline Rearranger
Binary Insertion of Textline	Perform Full-Text Search
Bring ToolBook Window to Front	Pie Chart
Bring up the Command Window	Play clip in a loop
ButtonStillDown Page Navigation	Play clip on enter page
Calculator	Playing MIDI Directly
Calendar	Playing random sounds
Can't Touch This	QWERTY Keyboard
Changing the behavior of BACK	Ratchet
Check for Alpha Non-Numeric Data	Recording wave files from CD
Check for State Abbreviation	Remove given char from string
Content Buttons	Remove trailing spaces
Controlling volume of clips	Scale Slider
Convert textlines to list	Search and replace in a string
Converting to Binary, Hex, Octal	Search Path for File
Create Full-Text Search Index	Searching Field
Dial	Self-sorting Field
Directories and Files	Show all objects of page
Disabling task switching	Showing a web page
Display 2-D array values	Show-off
Drag a Textline	Simple Bar Chart
Drag and Drop Text	Simple Line Chart
Dragging Objects	Slider
Dragging objects into bins	Sliders
Ex Select Field	Slow-motion
Exit and restart Windows	Slow-motion
Exploding Pie	Sorting a 1-D array
Get Current Path Setting	Sorting a 2-D array
Get day of week from date	Spin controls
Get Free Disk Space on a Drive	Tab Widget
Get number of days in month	Time and Date
Get path of current book	Timer
Get Printer Resolution	True and False
Initialize Excel Conversation	Useful Patterns
Insert/Overwrite field	Using a RECT structure
International Time and Date	Using system sounds
isLeapYear()	Using windows pointers
itemOffset()	Validate Numerical Data
Labels	Validate Text
Limit Length of Entry String	Windows and DOS versions
Making mmSource-paths book relative	Windows or WfW

Features of this sample application:



Navigation viewer that extends to allow limited editing of the content at Author level.

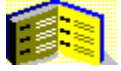




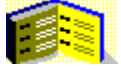
MAPI.TBK

The MAPI (Messaging Application Programming Interface) Tutorial will only work on machines that have MAPI compatible mail systems such as Microsoft Mail. The script for this book contains OpenScript wrapper functions that access the functionality of this DLL with easy to understand names like `getMsgText()`, which gets the text of a message, and `getMsgDate()`, which gets the date of a message. All reusable scripts are conveniently located in the book script.

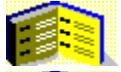
Features of this sample application:



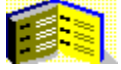
The MAPI DLL itself is linked in the `linkMAPI()` function.

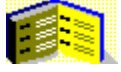


Other DLL links used by this book are performed in the `initLinks()` function.



Navigation is provided by a simple list-box menu and buttons on the background.

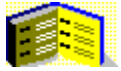




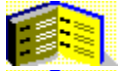
PRESENT.TBK

The present sample application is a very simple ToolBook application. The book flips pages and changes the color of text when it is clicked.

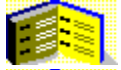
Features of this sample application:



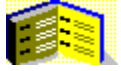
The application sizes itself to run full-screen.

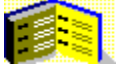


The menu page updates itself to reflect the current book.



The text (actually buttons) changes color to indicate the speaker's current topic





SCRAPBK.TBK

The scrapbook is a collection of graphics to be used as clip art in ToolBook applications. Every object in the scrapbook is made up entirely of ToolBook objects grouped together. The scrapbook uses a unique interface that allows users to drag and drop graphics from one ToolBook window to another. Because ToolBook objects cannot be pasted into other applications, these graphics can only be used in other ToolBook applications. You can easily add your own clip art to the collection by using the clipboard or copying it using drag-and-drop.

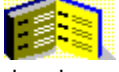
Features of this sample application:



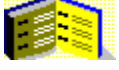
Allows users to print, add, or remove graphics from the Scrapbook.

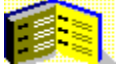


Associates user properties with each graphic so users can search for specific graphics.



Prints images using drag-and-drop printing. The drag detects if the Print Manager is beneath the drag image. No message needs to be sent to the Print Manager.





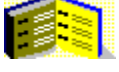
TAQUIN.TBK

A well-known game is used to demonstrate an unusual use of viewers. All the tiles in the Taquin game are viewers. There are enough viewers to accommodate the 5 by 5 grid. Handlers for moving the tiles are located at the book level. Any bitmap can be used for the puzzle and depending on the grid chosen, the viewers are divided up into different sizes.

Features of this sample application:

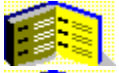


When a new graphic is chosen, it is imported onto page 2 as a paint object.



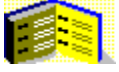
Each viewer displays a different portion of page two. This is accomplished by setting the `pageScroll` property of that viewer. This script, located in the book, performs the necessary math to determine the `pageScroll` of the viewer:

```
--SET PAGE_SCROLL FOR THE TILE (to show correct portion of puzzle)
x=(tCol-1)*tWidth
y=(tRow-1)*tHeight
set pageScroll of tileX to x*xPUPP,y*yPUPP
```



Popup help is provided.

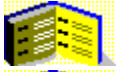




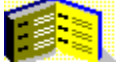
TMPLHELP.TBK

The template Help application provides help for the [CBT1.TBK](#) template. This is a good example of using ToolBook rather than Winhelp to provide context-sensitive help for an application. The book has a very simple design. There are four backgrounds, each of which has two record fields. The top record field contains a title. The bottom record field contains a detailed body of information about the topic. The four backgrounds in the book correspond to the four CBT tips books for which this book provides help.

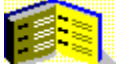
Features of this sample application:



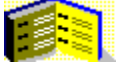
All scripts for this book are located in the book script.

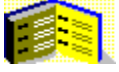


Hotwords are automatically linked to pages based on their name. Each page name is a combination of its background name (which corresponds to the name of the book to which it is linked) followed by an underscore (`_`) and a unique name. The first page of the background is a table of contents name and should have the same name as the background. For example, hotword "about" on background "cbt4" will link automatically to page "cbt4_about".



The template is customizable. To customize it, delete all but one page. Name the background after the book to which you want to link it, and name each page using the technique above.





WHOWHERE.TBK

The Who's Where book is a complete application using ToolBook as a front-end to a Paradox database. The entire Paradox engine is accessible via TB40PDX.DLL. This application makes full use of the Paradox engine. It uses a single table (WHOWHERE.DB), a primary index (WHOWHERE.PX), and a Binary Large Object (BLOB) file (WHOWHERE.MB). The BLOB file stores both memo and graphic BLOBs.

Features of this sample application:

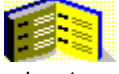


The script of this book includes OpenScript wrappers for important functions in the dll.

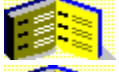


Who's Where demonstrates how to display bitmap and memo BLOBs from a Paradox database in

ToolBook.



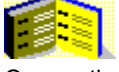
The application is network ready and will work with multiple users on a network by taking advantage of Paradox's record locking features.



The interface allows for dynamic creation and deletion of Paradox indexes.



Who's Where has two different modes: an Edit mode and a Browse mode. In Edit mode, data can be updated and records can be added or deleted. In Browse mode, the data is locked and cannot be changed.



Along with navigation buttons, other methods are provided for navigating through the database. One method allows users to search for names using an index and a map with the location of every employee's office.

