Custom Controls for Microsoft® Access

Total Access Components™

For Microsoft® Office Access

FMS

www.fmsinc.com

License Agreement

PLEASE READ THE FMS SOFTWARE LICENSE AGREEMENT. YOU MUST AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT BEFORE YOU CAN INSTALL OR USE THE SOFTWARE.

IF YOU DO NOT ACCEPT THE TERMS OF THE LICENSE AGREEMENT FOR THIS OR ANY FMS SOFTWARE PRODUCT, YOU MAY NOT INSTALL OR USE THE SOFTWARE. YOU SHOULD PROMPTLY RETURN ANY FMS SOFTWARE PRODUCT FOR WHICH YOU ARE UNWILLING OR UNABLE TO AGREE TO THE TERMS OF THE FMS SOFTWARE LICENSE AGREEMENT FOR A REFUND OF THE PURCHASE PRICE.

Ownership of the Software

The enclosed software program (“SOFTWARE”) and the accompanying written materials are owned by FMS, Inc. or its suppliers and are protected by United States copyright laws, by laws of other nations, and by international treaties. You must treat the SOFTWARE like any other copyrighted material except that you may make one copy of the SOFTWARE solely for backup or archival purpose, and you may transfer the SOFTWARE to a permanent storage device.

Grant of License

The SOFTWARE is available on a per license basis. Licenses are granted on a PER USER basis. For each license, one designated person can use the SOFTWARE on one computer at a time.

Use and Redistribution Rights

FMS grants you, the licensed developer, the right to distribute the SOFTWARE custom control files (those with a file extension of *.OCX) on a non-exclusive, royalty free basis, provided that:

1. Each developer using the program owns a license of the SOFTWARE.
2. You do not distribute any of the SOFTWARE’S License files (those files with a file extension of *.LIC).

3. You do not distribute the custom control files, in whole or in part, in any form that does not add primary and substantial value to the custom controls.
4. You agree to indemnify, hold harmless, and defend FMS and its suppliers or contractors from and against any claims or lawsuits, including attorneys' fees, that arise or result from the use or distribution of or other activities relating to your software.
5. If you claim a copyright, you must add a clause stating "Portions of this program are Copyright Total Access Components from FMS, Inc." and that you do not claim ownership of the SOFTWARE.
6. You do not permit (either by explicit or implicit means) further distribution of the SOFTWARE by your end users.

Restrictions

Under no circumstances may you attempt to reverse engineer this product. You may not rent or lease the Software, but you may transfer the Software and the accompanying written materials on a permanent basis provided you retain no copies and the recipient agrees to the terms in this Software License. Ownership transfers must be reported to FMS, Inc. in writing and are not accepted if the original developer already distributed applications using the Software.

Transfer of License

If your SOFTWARE is marked "NOT FOR RESALE", you may not sell or resell the SOFTWARE, nor may you transfer the FMS Software license.

If your SOFTWARE is not marked "NOT FOR RESALE", you may transfer your license of the SOFTWARE to another user or entity provided that:

1. You have not distributed applications including the SOFTWARE.
2. The recipient agrees to all terms of the FMS Software License Agreement.
3. You provide all original materials including software disks or compact disks, and any other part of the SOFTWARE's physical distribution to the recipient.
4. You remove all installations of the SOFTWARE.
5. You notify FMS, in writing, of the ownership transfer.

Limited Warranty

If you discover physical defects in the media on which this software is distributed, or in the related manual, FMS, Inc. will replace the media or manual at no charge to you, provided you return the item(s) within 60 days after purchase.

ALL IMPLIED WARRANTIES ON THE MEDIA AND MANUAL, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE LIMITED TO SIXTY (60) DAYS FROM THE DATE OF PURCHASE OF THIS PRODUCT.

Although FMS, Inc. has tested this program and reviewed the documentation, FMS, Inc. makes no warranty or representation, either expressed or implied, with respect to this software, its quality, performance, merchantability, or fitness for a particular purpose. As a result, this software is licensed “AS-IS”, and you are assuming the entire risk as to its quality and performance. IN NO EVENT WILL FMS, INC. BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE, OR INABILITY TO USE THIS SOFTWARE OR ITS DOCUMENTATION.

THE WARRANTY AND REMEDIES SET FORTH IN THIS LIMITED WARRANTY ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESSED OR IMPLIED.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitations or exclusions may not apply to you. This warranty gives you specific legal rights; you may also have other rights that vary from state to state.

U.S. Government Restricted Rights

The SOFTWARE and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable.

Manufacturer is FMS Inc., Vienna, Virginia. Printed in the USA.

Acknowledgments

We would like to thank everyone who contributed to make Total Access Components a reality. Thanks to the many existing users who provided valuable feedback and suggestions, and to all of our beta testers for their diligence and feedback.

Many people at FMS contributed to the creation of Total Access Components, including:

- **Product Design and Development:** Luke Chung, Scott Ellis, and Jim Ferguson
- **Quality Assurance and Technical Support:** Molly Pell, John Litchfield, Aparna Pophale, and Madhuja Vasudevan
- **Documentation:** Luke Chung, Scott Ellis, Molly Pell, and Aparna Pophale

Table of Contents

Welcome to Total Access Components!

Thank you for selecting Total Access Components, the only collection of ActiveX controls designed specifically for Microsoft Access. This is the seventh major version of Total Access Components since its debut with Access 2.0 and includes support for 64-bit environments.

Total Access Components is developed by FMS, the world's leading developer of Microsoft Access products. In addition to Total Access Components, we offer a wide range of products for Microsoft Access developers, administrators, and users:

- EzUpData (cloud hosting of Access reports and files)
- Total Access Analyzer (database documentation)
- Total Access Admin (database maintenance control)
- Total Access Detective (difference detector)
- Total Access Emailer (email blaster)
- Total Access Memo (rich text format memo fields)
- Total Access Speller (spell checker)
- Total Access Statistics (statistical analysis program)
- Total Access Startup (version launcher)
- Total Visual Agent (database maintenance and scheduling)
- Total Visual CodeTools (code builders and managers)
- Total Visual SourceBook (code library)

We also offer products and services for SQL Server, Visual Studio .NET, Visual Basic 6, and data analysts. Visit our web sites, www.fmsinc.com, and www.fmsasg.com, for more information. If you didn't purchase directly from us, please register online at www.fmsinc.com/support, and sign up for our free email newsletter. This guarantees that you are contacted in the event of news, upgrades, and beta invitations. Once again, thank you for selecting Total Access Components.

FMS

Luke Chung
President

Chapter 1: Introduction

ActiveX controls (originally called OLE custom controls and OCXs) were first introduced in Microsoft Access 2.0. They were also introduced for Visual Basic and quickly became a popular standard for enhancing forms and simplifying programming. Unfortunately, most ActiveX controls are designed for Visual Basic and do not work properly in Access forms and reports. Total Access Components addresses this problem by providing 30 professional controls specifically designed for Microsoft Access. This chapter introduces Total Access Components and its ActiveX controls, and provides an outline of the manual.

Topics in this Chapter

- **Introduction to Custom Controls**
- **Product Highlights**
- **Enhancements in This Version**
- **Enhancements from Previous Versions**
- **Visit Our Web Site**

Introduction to Custom Controls

If you have used Access forms and reports, you are already familiar with using built-in controls (e.g. text boxes, labels, combo boxes). Total Access Components provides a set of custom controls to enhance this list.

Control Types

A control is a generic term for a self-contained object that can be placed on an Access form or report. There are several types of controls:

Input Custom Controls

Allow users alternative ways to enter values by pressing, pulling, typing, dragging, etc. An example is a spin button to increment a number or date.

Output Custom Controls

Used to display values specified by the programmer, or to add visual elements to a form or report. Examples include gauges and progress meters.

Input and Output Custom Controls

Allow users to manipulate the control to change a value, or if the value is changed programmatically, the control's appearance reflects the new value. An example of this type of control is a slider bar.

Invisible Controls

Have no visible interface at all, but are used to provide other useful functionality. For instance, a control may be used to encapsulate Windows API services such as the common file dialogs or the Windows clipboard.

Product Highlights

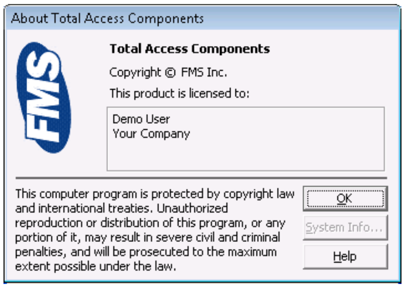
Total Access Components provides 30 controls that you can place on your forms and reports to enhance your application's appearance and functionality. Many controls allow you to provide new and enhanced display effects, while others allow you to use properties and methods to perform complex operations.

Document, Form, and Report Controls

These controls enhance the appearance of your forms, and reports:

About Box

Display information about your program in an easily customized dialog box. You can optionally supply a help file and access to the Windows System Information program if it is available on your user's machine.



About Total Access Components

Total Access Components

Copyright © FMS Inc.

This product is licensed to:

Demo User
Your Company

This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

About Box Control

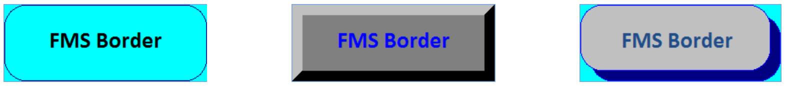
Bitmap Effects

Create catchy effects, fades, and transitions between bitmaps.

Bitmap Effects Control

Border

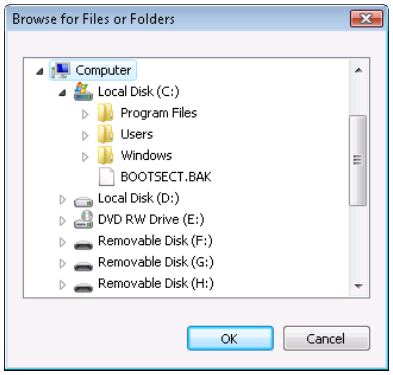
Create rectangular borders with a variety of styles: standard, three-D, shadowed, and rounded.



Border Control

Browse for Folder

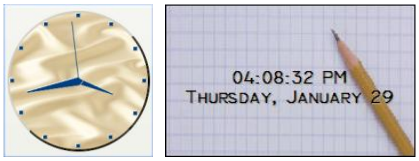
Display a browse for folder dialog, and specify whether your users can select a folder, a computer, a printer, or a file.



Browse for Folder Control

Clock

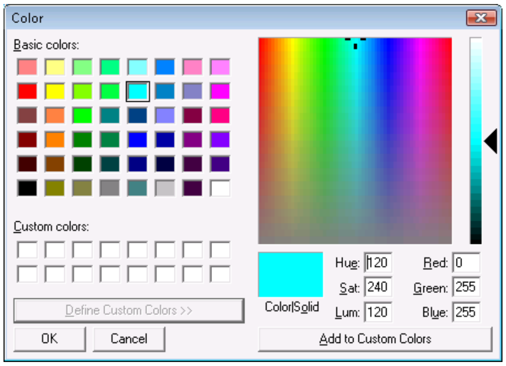
Display a graphical analog or digital clock in a variety of styles, with optional background pictures.



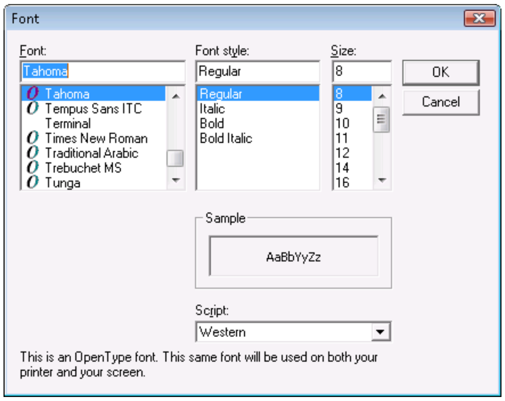
Clock Control

Common Dialog

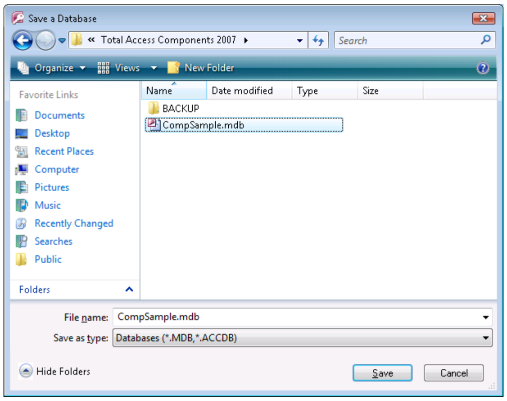
Use the Windows common dialog controls from your application. Easily add File Open/Save dialogs, Font and Color choosers, Printer Setup, and customized Help support to your application.


Hue: 120 Sat: 240 Lum: 120
Red: 0 Green: 255 Blue: 255

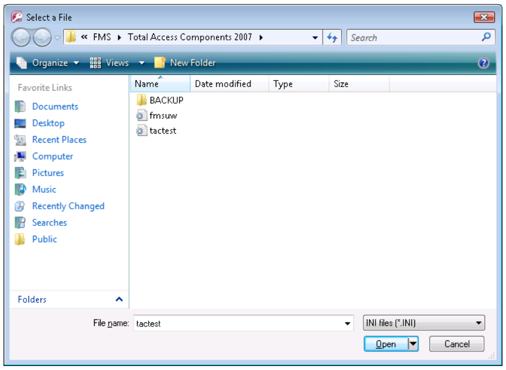
Common Dialog Control: Choose Color


Font: Tahoma
Font style: Regular
Size: 8
Script: Western
This is an OpenType font. This same font will be used on both your printer and your screen.

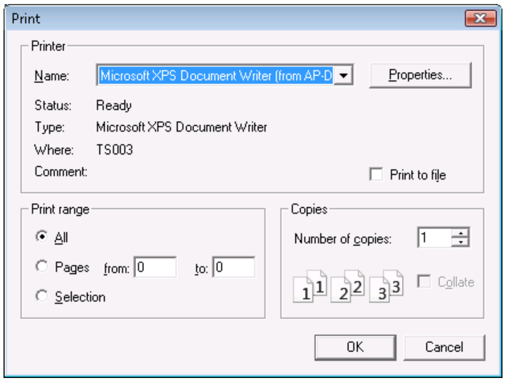
Common Dialog Control: Choose Font


« Total Access Components 2007 ›
File name: CompSample.mdb
Save as type: Databases (*.MDB,*.ACCDB)

Common Dialog Control: Save File


« FMS › Total Access Components 2007 ›
File name: tactest
INI files (*.INI)

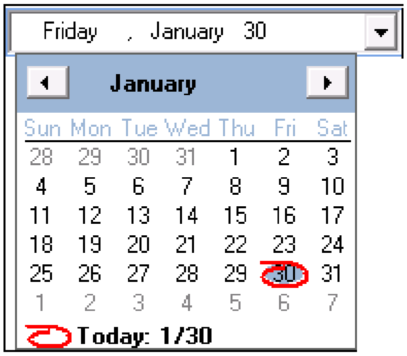
Common Dialog Control: Open File


Name: Microsoft XPS Document Writer (from AP-D
Status: Ready
Type: Microsoft XPS Document Writer
Where: TS003
Comment:
□ Print to file
Print range
● All
○ Pages from: 0 to: 0
○ Selection
Copies
Number of copies: 1
□ Collate

Common Dialog Control: Print

DateTime Picker

Allow users to view and select a date or time in the appropriate format, and optionally allow a drop-down calendar.



Friday , January 30

January

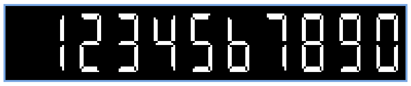
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Today: 1/30

DateTime Picker Control

Digital Display

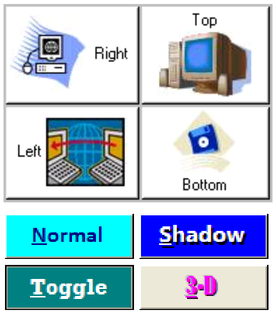
Display numbers using a digital display similar to those found on calculators.



Digital Display Control

Enhanced Button

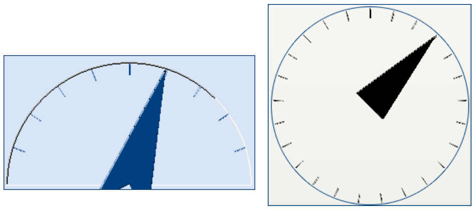
Go beyond the standard button control by combining a variety of text effects, animation, and bitmap styles.



Enhanced Button Control

Gauge

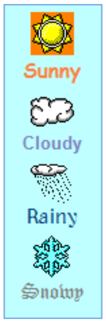
Simulate an analog gauge, such as a gas gauge or a pressure meter, to provide feedback to your users on the progress of an operation or the size of a value.



Gauge Control

Icon Menu

Add icon menus similar to the menu used in Outlook. Icon menus are a fancy way to organize and make selections.



Icon Menu Control

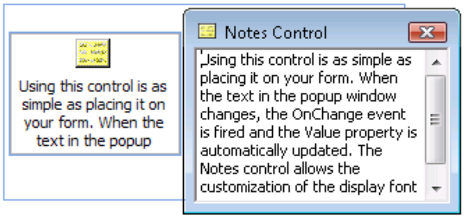
Marquee

Display text scrolling left, up, right, or down. This control supports multiple text styles.

Marquee Control

Notes

Use the Notes control to add popup notes to your form. Text can be bound directly to a field.


Using this control is as simple as placing it on your form. When the text in the popup

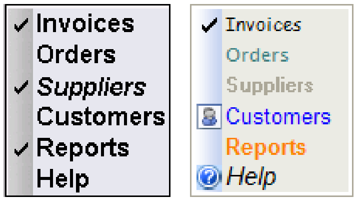
Notes Control

Using this control is as simple as placing it on your form. When the text in the popup window changes, the OnChange event is fired and the Value property is automatically updated. The Notes control allows the customization of the display font

Notes Control

Popup Menu

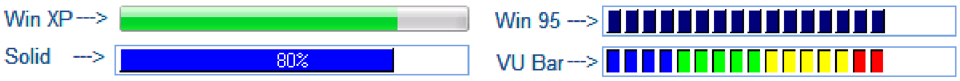
Add customized menu support to your forms. You can specify a popup menu to appear when a user right-clicks on objects in your application.



Popup Menu Control

Progress Meter

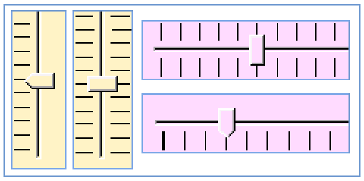
Provide feedback to your users with standard or segmented progress meters in a variety of styles. Choose from flat, Windows 95, or the unique segmented meter styles.



Progress Meter Control

Slider

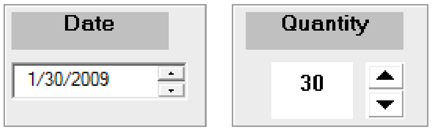
Add horizontal or vertical slider bars for user input or showing values.



Slider Control

Spin Button

Add horizontal or vertical spin buttons to allow your user to increase or decrease values. Press the up/down or left/right buttons of this control to change the values.



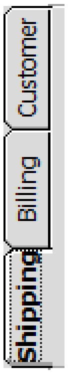
Spin Button Control

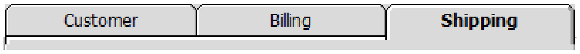
Splitter

Use a vertical or horizontal splitter to allow users to change the relative size of two parts of your Access forms

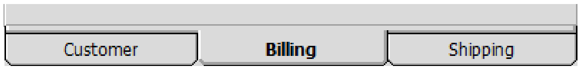
Splitter Control

Tab





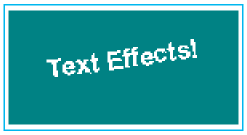
Create applications using the popular tabbed-dialog interface with support for different styles and positions (top, bottom, left, right).



Tab Strip Control

Text Effects

Display rotated text with custom display styles including three-D and drop-shadows.



Text Effects Control

Multimedia Controls

These controls add exciting multimedia effects to your database:

AVI

Play digitized video .AVI files through your computer.

CD Player

Play audio Compact Discs through your computer’s CD Player.

Wave and MIDI

Play Windows digitized audio .WAV files, and MIDI sequences through your computer’s sound card.

Programmer Utilities

The Utility controls simplify advanced programming tasks:

Cursor

The Cursor control sets the windows cursor to a variety of styles, including a custom image.

Clipboard

Easily add cut, copy, and paste features.

INI File

Read from and write to a Windows initialization file without using Windows API calls.

Registry

Read from and Write to the Windows Registry without using Windows API calls.

Resize

Automatically resize controls when your form is resized.

System Information

Retrieve information about your user’s computer, including free memory and disk space, and other hardware information.

Timer

Add multiple timers to a form.

Enhancements in This Version

The latest version of Total Access Components introduces support for Microsoft Access 2010, while continuing to support Access 2000 through 2007.

New 64 bit Version to Support Microsoft Access 2010

Total Access Components 2010 includes two programs in one to support the 32 and 64 bit versions of Microsoft Office 2010. The 64-bit version is a completely new set of controls with native support for 64-bit Access.

Microsoft does not allow both versions of Office 2010 to be installed on one machine. The Total Access Components 2010 setup program detects the appropriate version and installs it on your machine.

System Information Control

A new property, FreeMemEx, is available for the System Information control to support larger values when returning the system's total free memory.

A new Constant value, TacmSys8664, is available for the System Information control's CPUType property. This value is returned for 64-bit processors.

Enhanced Sample Database

The Sample Database is significantly enhanced to demonstrate all the properties, methods, and events for each control.

Enhancements from Previous Versions

If you are upgrading from Total Access Components 2003 or earlier, these features were added in the 2007 version and are included in this version.

Total Access Components 2007, version 12.0

New Cursor Control

Total Access Components includes a new Cursor control that allows you to specify a custom image to use as the windows cursor.

Control Enhancements

Many of the controls are enhanced with a more modern appearance and added functionality to support the features and look of Windows XP and Vista. These include:

- The **Clipboard Control** is enhanced to support copying and pasting bitmaps, in addition to text.
- The **Clock Control** is enhanced with support for background pictures, additional hand styles, the ability to hide hour ticks, support for Windows XP styles/themes, and smoother lines for a sleeker appearance
- The **Common Dialog Control** supports Compressed HTML Help (CHM) files in addition to previous HLP help files.
- The **Enhanced Button Control** includes a new option to display the standard cursor, or a Hyperlink Hand cursor when hovering on the button.
- The **Gauge Control** is enhanced with support for background pictures.
- The **Icon Menu Control** is enhanced with support for word wrapping on button text, customizable fonts and colors, and support for Windows XP styles/themes.
- The **Marquee Control** supports Windows XP styles/themes and word wrapping.
- The **Notes Control** supports Windows XP styles/themes and a variety of text locations.
- The **Popup Menu Control** supports Windows XP styles/themes, and allows customization of each menu item’s icon, font, and color properties.
- The **Progress Meter Control** supports Windows XP styles/themes, and allows larger values.
- The **Text Effects Control** supports word wrapping, options for text location, and Windows XP styles/themes.
- The **Timer Control** supports larger Interval values.

Visit Our Web Site

FMS is constantly developing new and better developer solutions. Total Access Components is part of our complete line of products designed specifically for the Access developer. Please take a moment to visit us online at **www.fmsinc.com** to find out about new products and updates.

Product Announcements and Press Releases

Read the latest information on new products, new versions, and future products. Press releases are available the same day they are sent to the press. Sign up in our Feedback section to have press releases automatically sent to you via email.

Product Descriptions and Demos

Detailed descriptions for all of our products are available. Each product has its own page with information about features and capabilities. Demo versions for most of our products are also available.

Product Registration

Register your copy of Total Access Components on-line. Be sure to select the email notification option so you can be contacted when updates are available or news is released. You must be registered to receive technical support.

Product Updates

FMS is committed to quality software. When we find problems in our products, we fix them and post the new builds on our web site. Check our Product Updates page in the Technical Support area for the latest build.

Technical Papers, Tips and Tricks

FMS personnel often speak at conferences and write magazine articles, papers, and books. Copies and portions of this information are available to you online. Learn about our latest ideas and tricks for developing more effectively.

Social Media

Join us on Facebook: http://www.facebook.com/fms.solutions

And follow us on Twitter: http://www.twitter.com/fmsinc

Newsgroups

Share your experiences, learn from others, and ask your questions in our virtual community. Visit our newsgroups at:

```
www.fmsinc.com/support/newsgrp.htm
```

Or see our web site for additional instructions.

Links to Other Development Sites

Jump to other locations, including newsgroups, user group home pages, and other sites with news, techniques, and related services.

Chapter 2: Installation

Total Access Components comes with an automated setup program to get you up and running as quickly as possible. This chapter describes the system requirements, installation steps, instructions for upgrading from previous versions, and instructions for uninstalling.

Topics in this Chapter

- **System Requirements**
- **Licensing Rules**
- **Installing Total Access Components**
- **Using the Update Wizard**
- **Upgrading from Previous Versions**
- **Uninstalling Total Access Components**

System Requirements

The system requirements for Total Access Components are similar to the requirements of Microsoft Access, and include:

- A copy of the Microsoft Access.
- Hardware and operating system supported by Microsoft Access.
- 10 MB free disk space to install the product.

Licensing Rules

Total Access Components is licensed on a *per developer* basis. Each developer must own a license. The standard Total Access Components package includes one license for one developer. FMS also offers quantity discounts and site license programs to let you economically add user counts.

The Total Access Components custom control file (TACOMP.OCX) may be distributed with your application to other users—those users do not need to own Total Access Components to use the controls within the context of your application. However, you ***may not*** distribute any license files (*.LIC) with your applications.

For complete information on licensing and restrictions, read the License Agreement at the beginning of the manual, and review **Chapter 4: Distributing Controls**.

Installing Total Access Components

Total Access Components is installed using an automated setup program. To install Total Access Components, follow these steps:

1. Locate and run the setup program.
2. When prompted, enter your registration information and product key (serial number).
3. Specify the destination folder for the files.
4. Be sure to read the readme file for any late breaking news that is not included in the manual or help file.

List of Files Installed

The INSTALL.LOG file in the Total Access Components installation folder lists the files installed plus a complete log of the Total Access Components setup, including overwritten files, and the registry entries and Start Menu items created.

Using the Update Wizard

If you are registered, you should receive emails from FMS when updates are released for the products. To verify you have the latest build, you can use the Total Access Components Update Wizard with an active Internet connection.

From the main Windows menu, select All Programs, FMS, Total Access Components, Update Wizard. Follow the prompts on the form to check for the latest update.

Upgrading from Previous Versions

If you are using an earlier version of Total Access Components, installing this version updates all Total Access Components controls automatically.

If you have distributed applications using a previous version of Total Access Components controls, you should re-deploy using the new runtime distribution program (TACRUN14.EXE) to ensure that users have the updated controls.

Tip

Since this version of Total Access Components supports Microsoft Access 2000 through 2010, previous installations of Total Access Components are no longer necessary and can be uninstalled.

Uninstalling Total Access Components

Total Access Components conforms to Windows installation and removal standards. To uninstall Total Access Components:

1. From the main Windows menu, select Control Panel.

2. From the Control Panel window, select Programs and Features (or Add/Remove Programs for Windows XP or earlier).
3. Select Total Access Components from the list of installed programs, and click the Change/Remove button.
4. Follow the onscreen prompts to uninstall the product.

Note that the uninstall process does not remove any files created by Total Access Components after it was installed. For example, a temporary database may still exist in the Total Access Components folder or working folder under your Users profile. After uninstalling the product, check these folders for any remaining files that need to be manually deleted.

Additionally, uninstalling does not delete Total Access Components controls from your forms and reports. You need to manually delete those.

Please keep in mind that the runtime license rules do not allow you to transfer your license to someone else if you have already distributed applications with Total Access Components.

Chapter 3: Using the Controls in Microsoft Access

Now that Total Access Components is installed, you may want to jump right in and start using the controls. Before you do, read the material in this chapter to understand important concepts in using custom controls. Even if you are experienced with using custom controls in products such as Visual Basic, you'll need to know some differences in the way custom controls work in Access.

Topics in this Chapter

- **Using the Sample Database**
- **Access Controls Vs. Custom Controls**
- **Using Custom Controls in Access Forms and Reports**
- **Using Custom Controls in VBA Code**
- **Changing to Pictures**
- **Custom Control Issues**

Using the Sample Database

One of the most important resources in Total Access Components is the sample database. This database is a standard Access database (.mdb) containing forms, reports, and code showing how to use every custom control included in Total Access Components.

The sample database is named **CompSample.mdb**, and is located in your installation folder. To open the database, click the Windows Start button and select Programs, FMS, Total Access Components, Sample Database.

In addition to providing examples of how to use the Total Access Components controls, the sample database is a tutor and guide for how to use the controls in your own databases. The forms, reports, and modules are designed to be easy to take apart so you can see how things work. None of the objects are secured, so you can work through the examples and see exactly how the custom controls works within an Access application.

Access Controls Vs. Custom Controls

In many respects, custom controls act like built-in Access controls. Like native controls, you can place custom controls on your form and manipulate them with the keyboard or the mouse. You can set their properties and interact with them at design-time and programmatically at run-time. Because they are not native to Access, however, there are some differences between custom controls and native controls.

Containership

Some standard Access controls act as containers for other controls. For example, an option group frame can contain option buttons. Custom controls, however, cannot act as containers. That is, you can't place controls "inside" a custom control. This is not a limitation of Total Access Components, but rather it is a limitation of custom controls in Access.

This makes using certain kinds of controls a little challenging. A Border control, for example, must simulate containership visually rather than acting as a container. You can simulate containership by converting a custom control to an image, as explained on 33.

Z-Order

Standard Access controls can partially or completely overlap each other. For example, a label control is often above the top edge of an option group control to create a caption for that control. When controls are stacked or overlapped, the order in which they appear is known as the Z-Order.

A limitation of custom controls in Microsoft Access is that controls cannot be placed on *top* of a custom control. Even if you position a control (such as a label) above a custom control in a form's design mode, Access hides that control when the form is in run mode.

One way to work around this limitation is to use the "Changing to Pictures" feature, discussed on page 33.

Control Visibility Differences

Some custom controls are never visible at run-time. These controls provide useful functions to programmers, but the end-user cannot see or interact with them. By contrast, all Access standard controls (with the exception of Page Break) are visible to the user.

Examples of invisible custom controls in Total Access Components are the Clipboard and Common File Dialog controls. These controls are accessible through Visual Basic for Applications (VBA) code, but are not visible on the form or report in run mode. The control is only visible in design mode and appears as an icon so that you can click on it and set its properties.

Using Custom Controls in Access Forms and Reports

To use custom controls in an Access application, the controls must be registered on your system. Installing Total Access Components automatically registers its controls, but when you create an application for distribution to your users, you need to register the controls on your users' machines. See **Chapter 4: Distributing Controls** for distribution information.

Tip

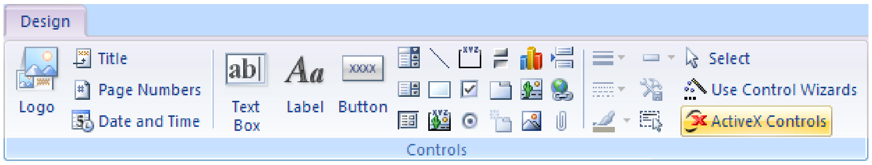
Please note that Total Access Components controls are not supported on Access Data Access Pages (DAPs). Microsoft deprecated DAPs in Access 2007.

Inserting Access Controls

Access built-in controls are displayed in the toolbox. You simply drag a control (e.g. text box, command button, etc.) from the toolbox and place it on your form or report.

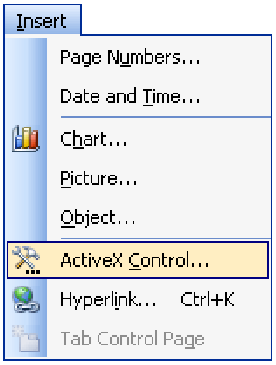
Inserting Custom Controls

Custom controls, such as the Total Access Components controls, are not available from the Access toolbox. Instead, they are available from the ActiveX Controls menu. To insert custom controls in Access 2010/2007, click "ActiveX Controls" on the Design ribbon:



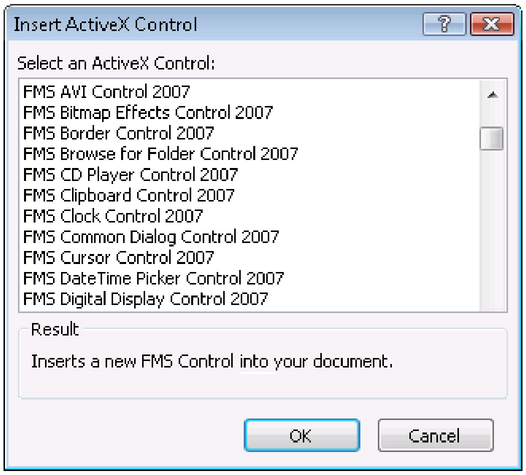
Inserting ActiveX Controls in from the Ribbon

In Access 2003 and earlier, select "ActiveX Control…" from the Insert menu:



Inserting ActiveX Controls in Access 2003 and Earlier

If Total Access Components is correctly installed, its controls are listed in the "Select an ActiveX Control" list:



Select an ActiveX Control:

FMS AVI Control 2007
FMS Bitmap Effects Control 2007
FMS Border Control 2007
FMS Browse for Folder Control 2007
FMS CD Player Control 2007
FMS Clipboard Control 2007
FMS Clock Control 2007
FMS Common Dialog Control 2007
FMS Cursor Control 2007
FMS DateTime Picker Control 2007
FMS Digital Display Control 2007

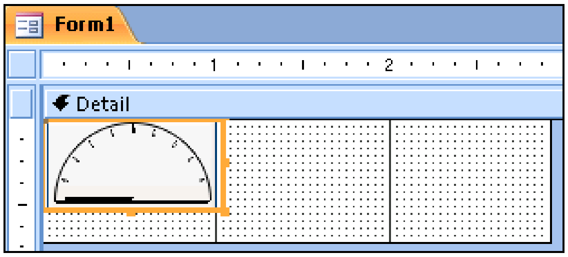
Result

Inserts a new FMS Control into your document.

Insert ActiveX Control Dialog

When selecting FMS controls, only select those with the prefix “FMS” and the suffix of “2010.” All ActiveX controls on your system are listed, including 16-bit controls, but only Total Access Components controls work properly in Access 2010, 2007, 2003, 2002, and 2000.

After selecting the control to insert, click [OK], and the control is inserted:



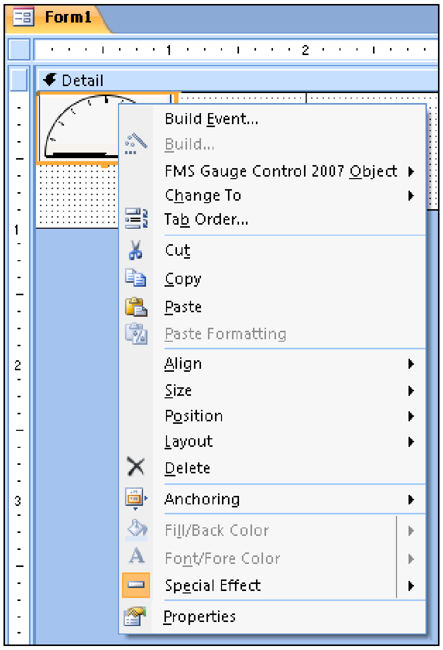
Control Inserted onto Form

Setting Control Properties Interactively

There are two methods for setting ActiveX control properties: the standard Access property sheet, and the ActiveX custom property sheet.

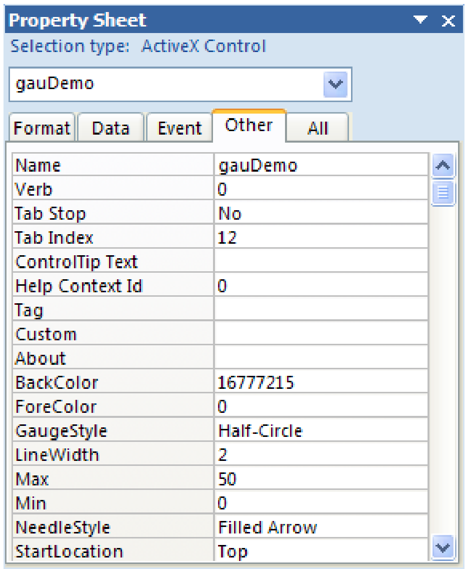
Using the Access Property Sheet

The Access property sheet can be used with any ActiveX control in Access. If the property sheet is not already open, right click on the ActiveX control, and select Properties from the popup menu.



Right-Click Context Menu

The Access property sheet displays both standard and custom properties. Set each property by editing the displayed value. For more information on a property, simply select the property and press [F1] for help.



Selection type: ActiveX Control

gauDemo

Name	gauDemo
Verb	0
Tab Stop	No
Tab Index	12
ControlTip Text	
Help Context Id	0
Tag	
Custom	
About	
BackColor	16777215
ForeColor	0
GaugeStyle	Half-Circle
LineWidth	2
Max	50
Min	0
NeedleStyle	Filled Arrow
StartLocation	Top

Standard Properties

Standard properties are those that Access automatically provides for all ActiveX controls. These standard properties are available in Access:

Property	Description
Visible	Determines if the control is visible
Display When	Determines when the control is visible
Left	Left position of the control
Top	Top position of the control
Width	Width of the control
Height	Height of the control
Special Effect	The effect applied to the control
BorderStyle	Appearance of a control’s border
BorderColor	Color of a control’s border
BorderWidth	Width of a control’s border
OLE Class	Description of the control object
Class	Class name of the control
Enabled	Enables or disables the User Interface for the control

If a control is databound, one additional property is available. For more information about Databinding, refer to page 26.

Property	Description
ControlSource	The field linked to the control's databound property.

Custom Properties

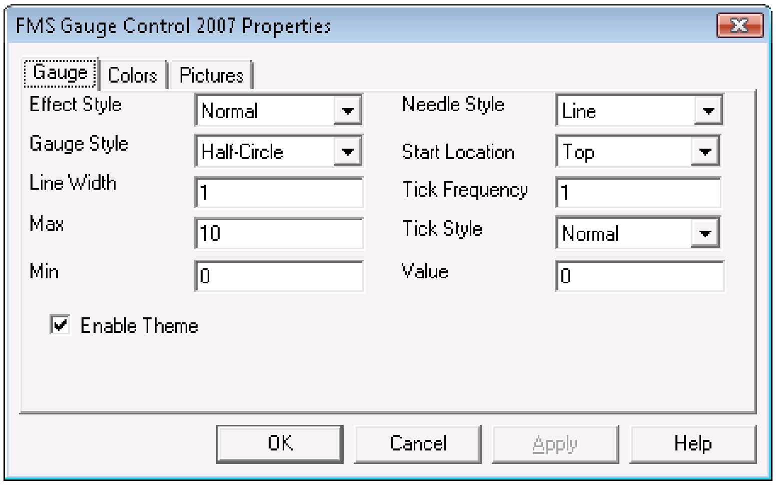
Custom properties are properties that are specific to the ActiveX control type. Custom properties can be found on the [Other] tab in the Access property sheet. For controls in Total Access Components, each custom property is explained in **Chapter 7: Property, Method, and Event Reference**.

Using the Custom Property Sheets

The Total Access Components control has a custom property sheet for specifying its unique settings. There are three ways to view the custom property sheet:

- Right click on the control, and select the FMS Total Access Components Control Object, Properties… menu item
- With the control selected, choose the Edit, FMS Total Access Components Control Object, Properties… menu
- Select the Custom property from the Other tab of the standard property sheet

The Properties menu item displays a customized property sheet that is specific to the custom control:



FMS Gauge Control 2007 Properties

Effect Style	Normal	Needle Style	Line
Gauge Style	Half-Circle	Start Location	Top
Line Width	1	Tick Frequency	1
Max	10	Tick Style	Normal
Min	0	Value	0

☑ Enable Theme

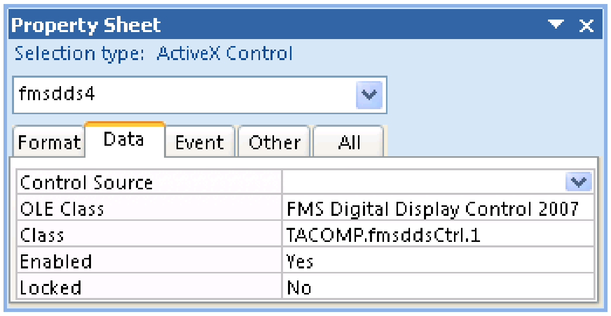
Custom Property Sheet

If you have a question on a setting or a value for a property, press the [Help] button or see **Chapter 5: Custom Control Reference**.

Databinding Custom Controls in Microsoft Access

Many of the Total Access Components controls support databinding, or linking a property of a control to a field in a table. If you change the value in the control, that value is automatically placed into the field, and if the value in the field changes, that value is reflected back to the control. This is accomplished without any code. In Total Access Components, the Value property can be bound for databound controls.

To databind a control, set the form's Record Source property to a table or query in the database. Then set the Control Source property of the control to a field in that table. The Control Source property can be set on the Data tab of the Access property sheet:



Selection type: ActiveX Control

fmsdds4

Control Source	
OLE Class	FMS Digital Display Control 2007
Class	TACOMP.fmsddsCtrl.1
Enabled	Yes
Locked	No

Access Control Source Property

Using Custom Controls in VBA Code

Some Total Access Components controls, such as the Text Effects and Bitmap Effects controls, can be useful without writing any code. Simply by changing settings in the property sheet, you can set static properties such as Caption, Style, Colors, etc. This may be sufficient for some applications.

However, to get the most out of the Total Access Components controls, you need to add a small amount of code. For the most part, this consists of setting properties and retrieving values, or telling the custom control to perform a method. Examples are included in the demo database and in **Chapter 5: Custom Control Reference**.

Tip

This manual assumes that you are familiar with adding events and setting properties using standard Access controls, and that you understand fundamental VBA programming concepts. For more information about VBA, consult your Microsoft Access documentation.. Examine the sample database and **Chapter 5: Custom Control Reference** for additional help.

Setting and Reading Properties

As described earlier, you can set properties for custom controls at design time in the Access property sheet. The Access property sheet makes no distinction between standard properties (which apply to all controls) and custom properties.

Similarly, when you set or retrieve a custom control's properties programmatically, there is no functional difference between standard and custom properties. For example, this code sets a standard property (Left) and a custom property (ClockStyle) for the FMS Clock control:

```
Me.clkDemo.Left = 1440
Me.clkDemo.ClockStyle = tacmClkAnalog
```

Likewise, when retrieving values, there is no difference between standard and custom properties:

```
intLeft = Me.clkDemo.Left
intClockStyle = Me.clkDemo.ClockStyle
```

Default Property

Some Total Access Components controls include a default property. If a control has a default property, you can refer to the control in code, and the default property is implied. For example, this both of these lines of code set the Value property of a Text Effects control:

```
Me.tefDemo.Value = "Rotated Text"
Me.tefDemo = "Rotated Text"       ' Value is implied
```

Setting Properties at Design Time vs. Code

Most properties can be set either at design time or programmatically at run time.

Why would you want to do one versus the other?

If a property doesn't need to change while the program is running, it is convenient to assign it in at design time in the property sheet. On the other hand, some people prefer seeing properties listed in their code so they don't have to remember how the properties were set in design mode. This is largely a matter of preference.

Some properties, however, are better set in code. For example, the FMS Common Dialog control allows you to specify the HelpFile property. The problem with setting this property in design time is that you must hard-code the path to the help file, or ensure that your help file is in a shared folder. If you set this property at run-time, you can dynamically determine where the file is located on a given machine:

```
Me.hlpDemo.HelpFile = strInstallFolder & "myapp.hlp"
Me.hlpDemo.HelpContext = Me.txtHelpID
Me.hlpDemo.helpcommand = TacmHlpHelpContext
```

Setting properties in code also lets you to make better use of Windows resources by reusing custom controls. For example, the same FMS Common Dialog control is used for the File, Font, Color, Print Setup, and Help functions. Instead of having five separate controls on your form, you can set the properties in code and display the appropriate style of dialog as needed.

Using Methods

Methods are special procedures or actions that let you customize the behavior of controls. Custom controls contain standard methods, and custom methods that are specific to the control type.

Perform an Action

Both standard and custom methods are invoked by specifying the control name, then the name of the method:

```
Me.nteDemo.SetFocus         ' standard method
Me.nteDemo.Show            ' custom method
```

Retrieve a Value

Some methods, like the FMS Clipboard control’s GetText method, are designed to return values to your program. This code uses the GetText method to read the contents of *clpClipBoard,* and to store the value to *txtTarget*:

```
txtTarget = Me.clpClipBoard.GetText
```

Use Arguments

Some methods accept “arguments” that control how the method works. For example, the FMS Registry control’s GetSubkeyString method accepts arguments for root key, key name, and key index:

```
Dim intCount As Integer
Dim intItem As Integer
Dim strValue As String
Dim strKey As String

strKey = "Word.document"

intCount =
Me.regWin.GetSubKeyCount(tacmReghKeyClassesRoot, _
                strKey)
If intCount > 0 Then
  For intItem = 0 To intCount - 1
    strValue = Me.regWin.GetSubKeyString( _
               tacmReghKeyClassesRoot, strKey, intItem)
    Debug.Print strValue
  Next intItem
End If
```

Standard Methods

Access automatically provides the following standard methods for all ActiveX controls:

Property	Description
Requery	Has no effect on Total Access Components custom controls
SetFocus	Sets the input focus to the control
SizeToFit	Has no effect on Total Access Components custom controls

Responding To Events

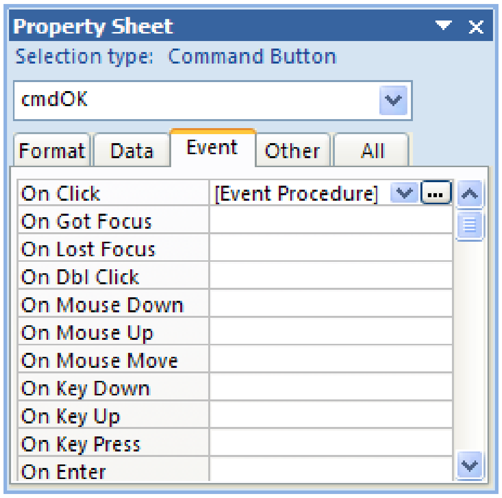
VBA programming is often called “event-oriented.” You do not write a top-down program that controls the flow of the application. Instead, you control the process by manipulating controls on your forms. In response to these user events, you write code to accomplish the necessary task.

These standard events apply to custom controls:

- OnUpdated
- OnEnter
- OnExit
- OnGotFocus
- OnLostFocus

In Access, you can attach a macro, a procedure in a standard module, or an “event procedure” to the event property of a control. Only event procedures apply to custom events in Total Access Components controls.

Attach code to specific events on standard Access controls by specifying the [Event Procedure] for the event.



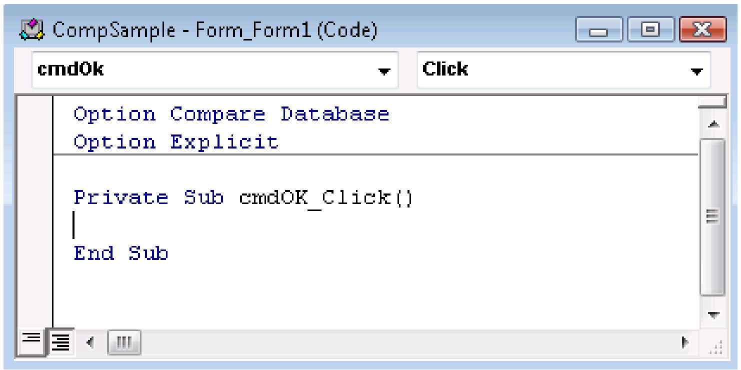
Selection type: Command Button

cmdOK

On Click	[Event Procedure]
On Got Focus	
On Lost Focus	
On Dbl Click	
On Mouse Down	
On Mouse Up	
On Mouse Move	
On Key Down	
On Key Up	
On Key Press	
On Enter	

Standard Event Properties

Click the […] button associated with the event property, and select “Event Procedure” from the Choose Builder dialog. Access automatically creates an event procedure stub in a module, and attaches it to the event property:



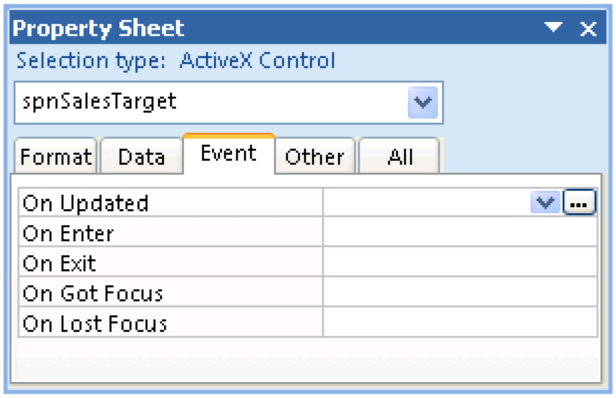
```
Option Compare Database
Option Explicit

Private Sub cmdOK_Click()

End Sub
```

Default Event Code for a Standard Control

Access generates the event when a user presses the button. It is up to you to add code in the event procedure to perform some action:

```
Sub cmdOK_Click()
  MsgBox "Exiting application"
  DoCmd.Quit
End Sub
```

Some (but not all) events for custom controls can be accessed in the same way as standard controls. The following event property sheet is from an FMS Total Access Components Spin control:



Property Sheet

Selection type: ActiveX Control

spnSalesTarget

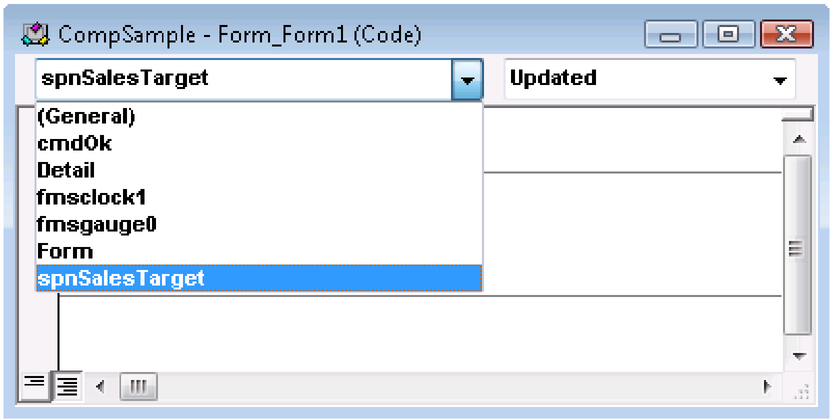
Custom Control Event Properties

For instance, you can use the standard OnGotFocus event to change the color of an associated text box:

```
Sub spnSalesTarget_GotFocus()
  Me.txtSales.BackColor = QBColor(1)
End Sub
```

However, most of the important custom events for Total Access Components controls are not in the standard Access event property sheet. For example, an important event for a Spin control is the OnChange event.

To create an event procedure for a Spin control's OnChange event, open the form's module and select the control from the drop-down list on the module toolbar:



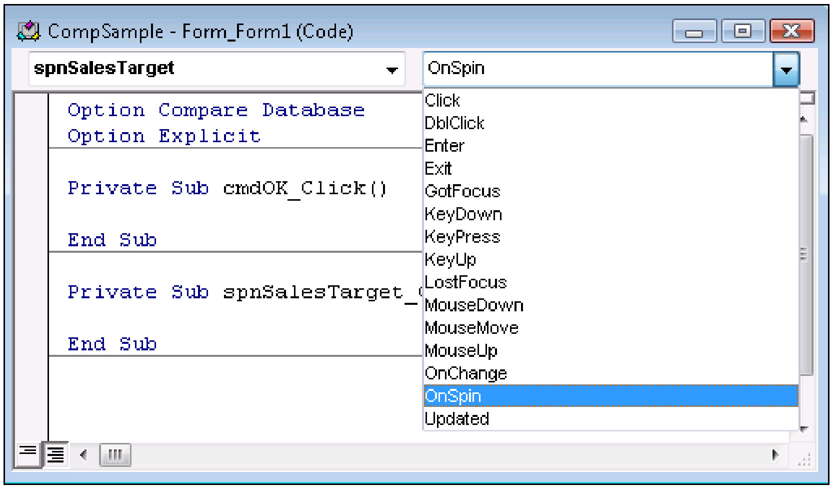
CompSample - Form_Form1 (Code)

spnSalesTarget

Updated

Select the Custom Control

Then select the event from the Event List:



```
Option Compare Database
Option Explicit

Private Sub cmdOK_Click()

End Sub

Private Sub spnSalesTarget_

End Sub
```

Select the Custom Event

If you did not previously create code for this custom event, Access inserts the stub code for you:

```
Sub spnSalesTarget_OnChange()

End Sub
```

Add the code to execute each time the event fires:

```
Sub spnSalesTarget_OnChange()
  Me.txtSales = Me.spnSalesTarget.value
End Sub
```

Some events, like the Spin Button’s OnSpin event, have additional information that is passed in the form of arguments.

In this example, the OnSpin event for “spnDiscount” is fired when the button is clicked. In addition to knowing when the button is clicked, you also know whether the user pressed the “up” or the “down” part of the Spin button:

```
Sub spnDiscount_OnSpin(Increased As Integer)
  If Increased Then
    Me.txtSales = Me.txtSales + 1
  Else
    Me.txtSales = Me.txtSales - 1
  End If
End Sub
```

If the user presses the “Up” button, the Increased argument is True (-1). If the user presses the “Down” button, the Increased argument is False (0).

VBA Error Handling

If you attempt to assign an invalid value to a custom control, Access generates a run-time error that halts the execution of your program. For this reason, it is a good idea to validate the values before assigning them to the control's properties:

```
Dim intMin As Integer
Dim intMax As Integer

intMin = Me.prgRating.Min
intMax = Me.prgRating.Max

If rating >= intMin And rating <= intMax Then
  Me.prgRating.Value = rating
Else
  MsgBox "Invalid Rating value: " & rating
End If
```

Alternatively, you could use VBA error handling if you anticipate that you might be assigning invalid data:

```
On Error Resume Next
Me.prgRating.Value = Rating
If Err <> 0 Then
  MsgBox "Error assigning Rating: " & Rating
End If
On Error GoTo 0
```

Some Total Access Components controls are designed to return information in the form of error codes. In order to use them correctly, you must trap for and test for specific error conditions. For example, the FMS Common Dialog control can be set to generate a run-time error if the user clicks [Cancel] on a File Open dialog. See **Error Handling** on page 92 for errors returned from specific controls.

See your Microsoft Access documentation for more information on error handling techniques.

Changing to Pictures

Microsoft Access offers the option to convert custom controls to images by right-clicking on the control, and choosing "Change To, Image".

When you change a custom control to an image, you are replacing the actual custom control with an image (or picture) of the control. The actual control on the form or report is no longer a Total Access Components Custom Control.

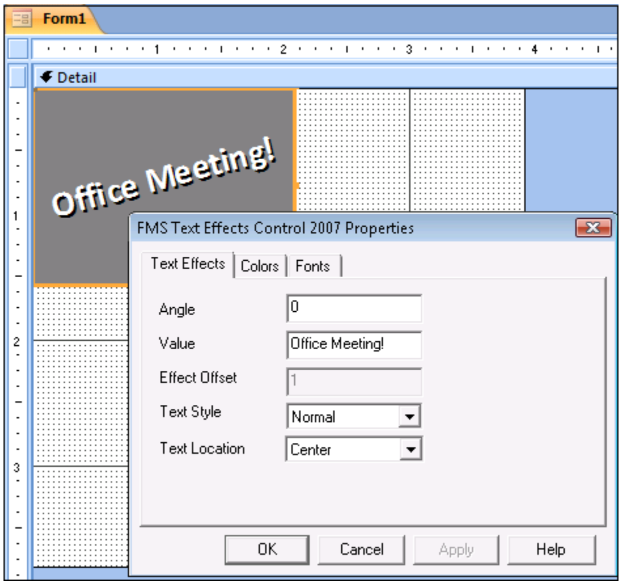
Since the object is no longer a custom control, but simply a picture of one, there are several things to consider:

- The object no longer receives custom events fired by the control. Standard Access events, such as OnGotFocus and OnClick, can still be used.
- Custom properties no longer exist and therefore can't be read or set. For example, you cannot get the "value" property of a *picture* of a gauge control.
- After you change a custom control to a picture, it is a static object and is unable to perform any custom methods.

When to Change

There are several reasons to change a custom control to an image. One common reason is to work around Access's limitations on custom controls. A custom control cannot be a container for another control, and you cannot place another control on top of a custom control. If you change the custom control to a picture, then you can place controls on top of it so that it looks like a container. For instance, you can add an FMS Border control to your form, adjust its properties, and then change it to an image in order to place other controls on top of it.

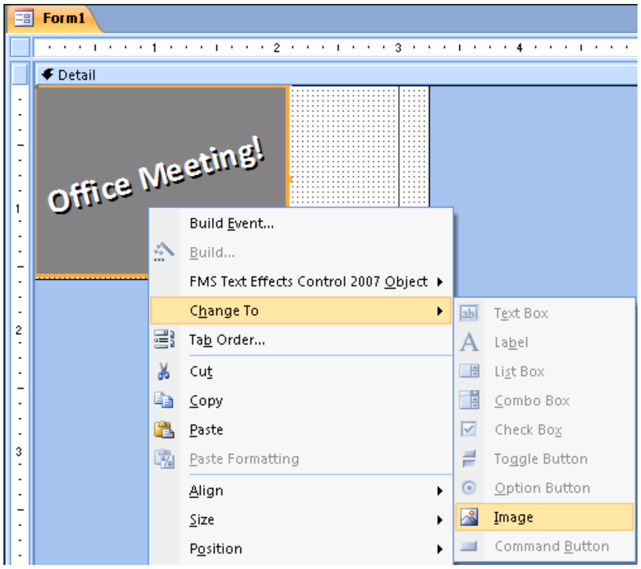
Another reason to change a custom control to an image is when it doesn't require interaction from the user. If you add an FMS Text Effects control to your form, you can set its caption and rotation angle in design mode:



Angle: 0
Value: Office Meeting!
Effect Offset: 1
Text Style: Normal
Text Location: Center

Adjusting Properties of a Text Effects Control

Once the design is set, there is no reason for it to remain a custom control. Unless you need to change the caption, rotation angle, or other property at run-time, convert it to an image:



Converting to an Image

Images use less memory than custom controls. If you have many custom controls on a form, you can reduce the form's memory requirements by changing them to images where appropriate.

When Not To Change

Some controls depend on user interaction. For example, the FMS Slider control allows users to manipulate its value by moving the slider. If you change this control to an image, it no longer functions.

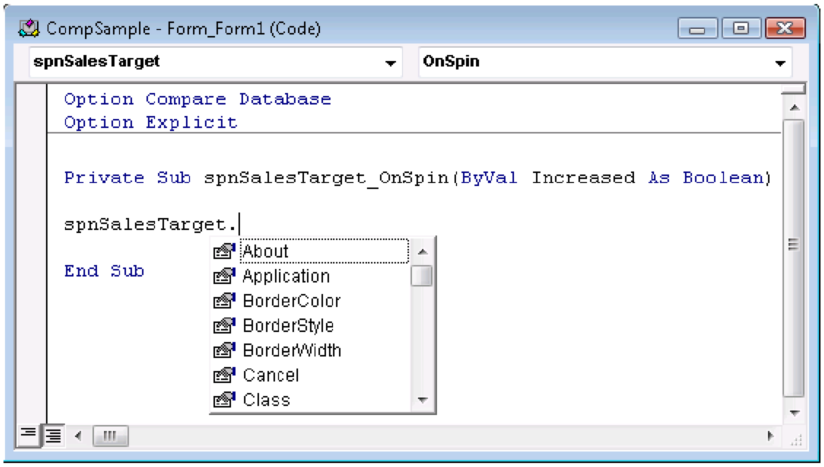
Additionally, if you need to reference properties programmatically or call methods for the control, don't change the control to an image. For example, if you need to change the rotation angle or caption of a Text Effects control at run time, don't change it to an image:

```
If Me.Status = "P" Then
  Me.tefDisplay.caption = "Company Picnic!"
  Me.tefDisplay.angle = 45
ElseIf Me.Status = "$" Then
  Me.tefDisplay.caption = "Payday!!"
  Me.tefDisplay.angle = 315
End If
```

Custom Control Issues

Complete Word Feature

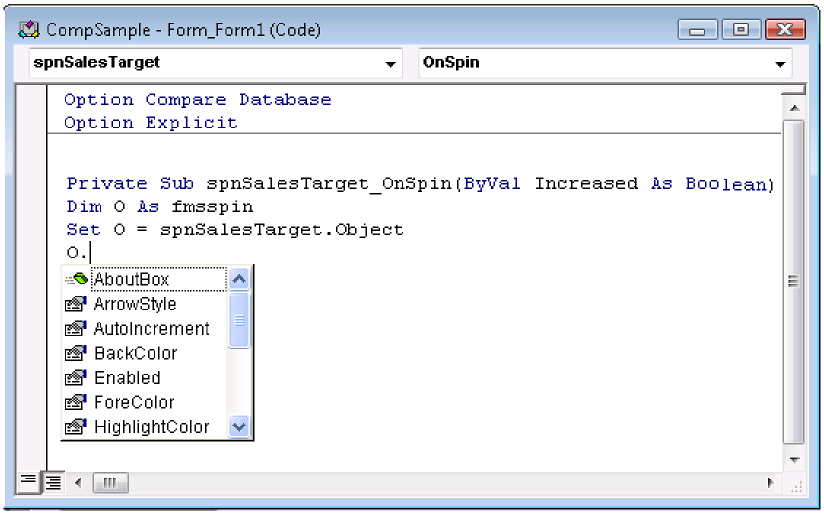
Microsoft Access does not fully implement the Complete Word feature for custom controls. The custom properties and methods of a control are not listed in the drop down. For example, when working with the FMS Spin Button Control, only standard properties and methods are listed:



```
Option Compare Database
Option Explicit

Private Sub spnSalesTarget_OnSpin(ByVal Increased As Boolean)

spnSalesTarget.

End Sub
```

Complete Word Feature

The custom properties are there, but they are not shown. However, you may work around this issue by declaring an object variable as the type of the control, then setting the variable to an instance of the control. For example, the following code enables the Complete Word feature for an FMS Spin Button Control:



```
Option Compare Database
Option Explicit

Private Sub spnSalesTarget_OnSpin(ByVal Increased As Boolean)
Dim O As fmsspin
Set O = spnSalesTarget.Object
O.
```

Using the Object Syntax

Default Font

The default font used in the custom controls is MS Sans Serif, 8 point.

Font Size Adjustments

For certain font sizes, fractional numbers are used. This behavior can be observed by opening the property sheet of a control with a Font property (i.e. the Text Effects control). On the Font property page, set the font size to 8, and press [Apply]. The font size is displayed as 8.25. Please note that the font displays in the proper size. This behavior is common to all ActiveX/OCX custom controls, and does not cause problems.

Attaching Labels

Several of the built-in Access controls are created with associated labels (for example, the text box, combo box, subform, and option group controls). When you move the control, the label goes with it.

In addition to identifying the function of the associated control, the label can be used to create a “hot-key” for the control so that the user can move to the control without using the mouse or pressing the [Tab] key.

When you add a custom control to a form however, an associated label control is not automatically created. To attach a label to a custom control, follow these steps:

1. Insert a new label onto your form or report and set its caption to the appropriate value. You can use the “&” character to create a hot key character.
2. Click on the label control once to select it. Then type Ctrl-X, or select Edit, Cut from the menu to cut the label onto the Windows clipboard. The label temporarily disappears from the form.
3. Click once on the custom control you wish to associate with the label to select it.
4. Type Ctrl-V, or select Edit, Paste from the menu to paste the label back onto the form. It is now associated with the custom control.

Chapter 4: Distributing Controls

Total Access Components includes a royalty-free runtime license so you can distribute the custom controls with your applications to non-Total Access Components owners. This chapter provides the details about distributing the Total Access Components custom controls with your Access databases.

Topics in this Chapter

- ➢ **Licensing Issues**
- ➢ **Single OCX File**
- ➢ **Simple Distribution**
- ➢ **Advanced Distribution**

Licensing Issues

There are several license terms with which you must comply to distribute Total Access Components controls with your applications.

The first and most important license issue is that you **may not distribute any Total Access Components license files** in any form. License files are the files installed by Total Access Components with the .LIC file extension. License files allow you, as the licensed developer, to place Total Access Components custom controls in your application. You are not licensed to redistribute Total Access Components custom controls in a way that allows your users to place them on their forms. If you do not distribute any license files, this is handled automatically—your users cannot interact with the controls except within the context of your application.

The second issue is that FMS does not provide technical support or customer service for your users—that it is up to you. We provide technical support to you as the developer, not to your users.

For complete licensing rules, please read the License Agreement at the front of the manual.

Single OCX File

Custom controls for other environments such as Visual Basic are usually distributed on a “1 custom control=1 OCX file” basis. Typically, only one or two custom controls are included in any given OCX file.

With Total Access Components, all controls are located in a single OCX file called TACOMP.OCX. We grouped all Total Access Components controls into a single OCX file to make the process of distributing the controls as easy and error-free as possible. This “one file” approach offers several benefits:

- Determining which ActiveX controls are used in an Access database is difficult. If you know you are using at least one Total Access Components control, simply distribute the TACOMP.OCX file and you are safe no matter how many Total Access Components controls you use.
- You only need to register one OCX file on your users’ computers.
- Updates are easier, since you only need to install one new file when the Total Access Components controls are upgraded.

As a result, Total Access Components controls are easy to distribute.

Simple Distribution

To make Access aware of Total Access Components, the controls must be registered on the machine that is running Access.

When you install Total Access Components on your development computer, the Setup program handles this automatically. When you distribute your application to other users, they do not have the Total Access Components Setup program. Therefore, you must configure your users' computers to use Total Access Components.

To distribute Total Access Components, use the runtime distribution program, **TACRUN14.EXE**, which is located in your installation folder. This program configures a computer to use Total Access Components in runtime mode. Simply execute TACRUN14.EXE on the target computer, and Total Access Components is automatically installed and configured.

Tip

TACRUN14.EXE can be executed with the optional parameter [/s] to run the distribution program silently, e.g.

```
    TACRun14.exe /s
```

There are three scenarios for using TACRUN14.EXE for distribution.

Access is Already Installed

If your user already has Microsoft Access installed on their computer, there are two steps to take:

Distribute Your Database(s)

Provide your users with a copy of your database by copying the database to their computer, or by placing the database on your network.

Execute TACRUN14.EXE on Their Computer

Next, you must execute TACRUN14.EXE on your user's computer. Total Access Components is now installed and configured.

Distribution with the Office 2010/2007 Package Solution Wizard, Office 2003 Package Wizard, or Office XP Packaging Wizard

If you are using the distribution tools available in the Office Developer Extensions/Developer Toolkit, follow these steps to ensure that Total Access Components is configured correctly:

1. In the list of dependencies, uncheck TACOMP.OCX. This file is distributed and registered when your setup runs TACRUN14.EXE.

2. In the list of included files, select “Add File” and choose TACRUN14.EXE (located in the Total Access Components installation folder).
3. On the Run on Complete page, check “Run this command when installation is finished,” and use the dropdown to select TACRUN14.EXE.

For more information, please refer to the documentation that accompanies the Developer Extensions or Developer Toolkit.

Using the Package and Deployment Wizard in the Microsoft Office 2000 Developer Toolkit

If you are using the Package and Deployment Wizard in Office 2000 Developer, follow these steps to ensure that Total Access Components is configured correctly:

- In the list of included files, uncheck TACOMP.OCX. This file is distributed and registered when your setup runs TACRUN14.EXE.
- In the list of included files, select “Add” and choose TACRUN14.EXE (located in the Total Access Components installation folder).
- On the Installation Options page, check “Run this command when installation is finished,” and type “TACRUN14.EXE” (without the quotes).

For more information, please refer to the documentation that accompanies Microsoft Office Developer 2000.

Advanced Distribution

Distributing Total Access Components without the TACRUN14.EXE program involves more work. This method is only recommended for advanced developers who understand the issues involved with ActiveX distribution and registration. If you are using the one of the previously described distribution methods, you do not need to read this section.

You must make sure your setup program copies and registers the Total Access Components control file: TACOMP.OCX. The following support files (with the same or higher version number) are also required:

Filename	Version Number
mfc42.dll	6.0.8665.0
msvcrt40.dll	4.2000.0.6201
MSSTKPRP.DLL	6.0.81.69
msvcirt.dll	6.1.8637.0

MSVCP60.DLL	6.0.8168.0
msvcrt.dll	6.1.8637.0

Registering Controls

If your installation program does not automatically support the registration of custom controls, you need to run a registration program.

Microsoft Windows includes a utility that supports the registration of custom controls. This program is called REGSVR32.EXE and can be found in the Windows or Windows\System folder. The REGSVR32.EXE program uses the following syntax:

```
REGSVR32.EXE [/u] [/s] controlname.OCX
```

where

```
/u - Unregisters the control
/s – Doesn’t display a message box
```

Location of Custom Controls

Although you can have your installation program copy the Total Access Components OCX file (TACOMP.OCX) to any folder on the user’s computer, we recommend that you copy it to the **Windows\System,** or **Winnt\System32** under Windows NT folder. This is the standard location for custom controls.

Files That You May Redistribute

The following files can be redistributed with your application royalty-free:

File Name	Description
mfc40.dll	Runtime support file
msvcrt40.dll	Runtime support file
Olepro32.dll	Runtime support file
TACOMP.OCX	Custom Control file containing all the Total Access Components controls
TACRUN14.EXE	Program to configure a computer to use Total Access Components in runtime mode

You may not redistribute any other files. Redistribution of any other Total Access Components file is a violation of your license agreement.

Chapter 5: Custom Control Reference

This section contains the reference material for each of the controls. Information on all properties, methods, and events is included here.

Topics in this Chapter

How to Use This Chapter

This chapter describes each control and lists its properties, methods, and events. Since many properties, methods, and events are identical for multiple controls, a detailed description of each is provided in **Chapter 7: Property, Method, and Event Reference**.

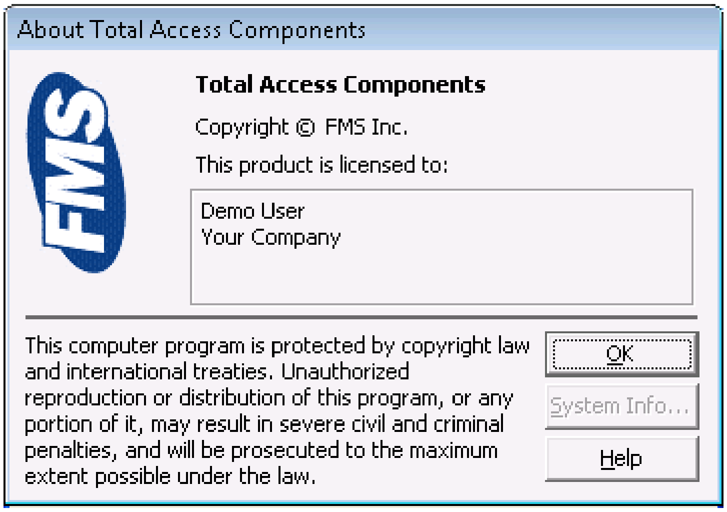
Tip

Default properties are indicated by the **Default** keyword. Databound properties are indicated by the **Databound** keyword.

For more information about default properties, see page 27, and for more information about databound properties, see page 26.

About Box

The About Box control displays information about your program in an easily customized dialog box. You can optionally supply a help file and access to the Windows System Information program if it is available on your users' machines.



About Total Access Components

Total Access Components

Copyright © FMS Inc.

This product is licensed to:

Demo User
Your Company

This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

About Box Control

Using the About Box

Insert the About Box control on your form, and then open the property sheet by right clicking on the control and selecting "Properties". Set the properties to values appropriate for your application.

Invoke the Show method when you wish to display the About Box.

About Box Properties

Name	Description
ApplicationName	The name of the application.
Copyright	The copyright text.
DialogTitle	The dialog title.
Font	The font attributes for the dialog.
HelpCaption	The caption for the [Help] button.
HelpCommand	The help action to perform when the user clicks [Help].
HelpContext	The Context ID for the help file.
HelpEnable	True to enable the [Help] button.
HelpFile	The file name of the help file.
HelpKey	The keyword that identifies the requested Help topic.
LicensedTo	Information about the licensed user (name, company, etc.).
LicenseString	The text to display in the copyright warning box.
Picture	The bitmap, metafile, or icon to display.

About Box Methods

Name	Description
LoadPicture	Loads a bitmap into the Picture property.
Show	Displays the dialog.

AVI (Audio Video)

The AVI control plays Windows digitized video *.AVI files. You can play, pause, stop, rewind, and navigate in these video files, and set repeat play.

Tip

Your machine must have a sound card installed and operating correctly in order for the sound track to be played.

Using the AVI Control

Insert an AVI control on your form, and name it appropriately. Then set the FileName property in the property sheet, or write code to set the FileName property at runtime.

Use the Close, Open, Pause, Play, Rewind, or Stop methods to control the file at runtime.

Stopping and Rewinding

The Stop method stops playing the file and rewinds to the beginning. If the Play method is called and the file plays until completion, the control also rewinds to the beginning.

AVI Properties

Name	Description
BackColor	The background color of the control.
FileName	The path and name of the AVI file.
Length	The play length in milliseconds.
Position	The current position in milliseconds.
Default	
Repeat	True to replay the file.
RepeatCount	How many times to repeat play (integer value).
Stretch	True to stretch the video to the size of the control.

AVI Methods

Name	Description
Close	Close the AVI resource.
Open	Open the AVI resource.
Pause	Pauses the AVI file.
Play	Plays the AVI file.
Rewind	Rewinds the AVI file to its beginning.
Stop	Stops the AVI file and rewinds.

AVI Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over the control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.
OnNewPosition	Fires when a new position in the movie is reached.
OnReplay	Fires when play starts over.

Bitmap Effects

The Bitmap Effects control lets you create catchy effects, fades, and transitions between bitmaps.

Bitmap Effects Control

Using Bitmap Effects

There are three basic steps for using the Bitmap Effects control:

1. Determine which two bitmaps to use in the animation.
2. Set the bitmap properties on the property sheet.
3. Call the animate method on your form, or set the CurrentStep property at runtime.

Tip

Bitmap Effects Tips

The Bitmap Effects control always stretches bitmaps to the size of the control. You should select bitmaps or size the control appropriately so that the bitmaps do not distort.

Your screen colors should be higher than the colors of the bitmaps you are displaying. If you are animating two 256-color bitmaps, your screen colors should be higher than this. If there are not enough colors, the bitmaps are not displayed properly

Bitmap Effects Properties

Name	Description
AnimationEffect	The animation effect to apply to the bitmaps.
CurrentPicture	The currently displayed picture.
CurrentStep **Default**	The current step in the animation process.
Delay	The delay, in milliseconds, between each step when using the Animate method.
Picture1	The first bitmap, metafile, or icon to display.
Picture2	The second bitmap, metafile, or icon to display.
StartPicture	The index of the picture to display initially when running the Animate method (either 1 or 2).
Steps	The number of steps in the animation. The more steps, the slower the transition between pictures.

Bitmap Effects Methods

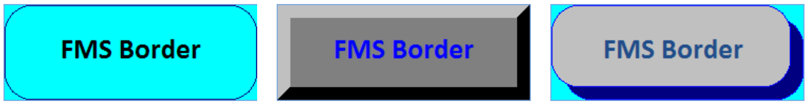
Name	Description
Animate	Plays the specified animation.
LoadPicture1	Loads a bitmap into the Picture1 property at run time.
LoadPicture2	Loads a bitmap into the Picture2 property at run time.

Bitmap Effects Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

Border

The Border control creates rectangular frames with a variety of styles, including standard, three-D, shadowed, and rounded.



Border Control

Using the Border Control

Follow these steps to use the Border control:

1. Place the control on the form, and size and position it appropriately.
2. Set the border properties on the property sheet.
3. Convert the control to an image by right-clicking on the control and selecting Change to, Image (see page 33 for details).
4. Set the Enabled property to False.
5. Select Format, Send to Back.

Change to Picture

Unless your application needs to programmatically change the Border control’s properties at runtime, you should use the Access “Change To Image” operation (see page 33 for details). This results in a more efficient form that loads faster and uses less memory.

When you change the Border control to a picture, the control uses the current Windows Color Depth setting. If your system is set to 256 colors, the Border control is converted to a 256-color bitmap, and won’t display properly on systems set to 16-color mode. Before converting controls to pictures, you should set your Windows Color Depth setting to the minimum color depth supported by your application.

Border Properties

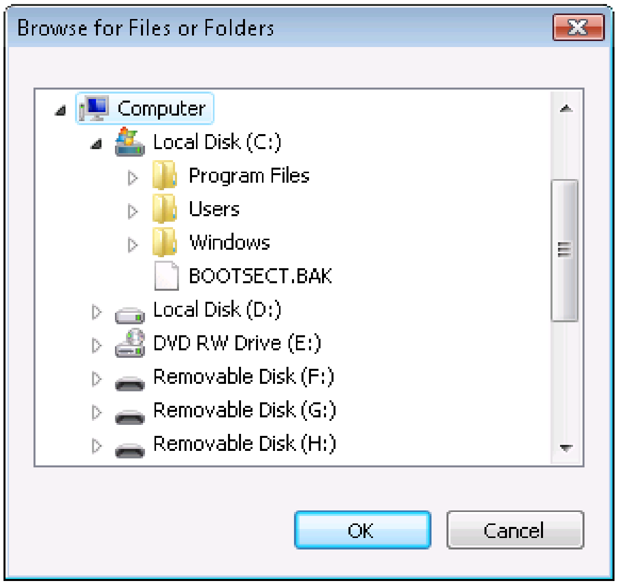
Name	Description
BackColor	The color of the bevel control’s background.
BevelWidth	The width of the bevel in pixels.
BorderColor	The color of the border.
CurveHeight	The amount of y curvature for rounded rectangle.
CurveWidth	The amount of x curvature for rounded rectangle.
EffectStyle **Default**	The style of the border.
HighlightColor	The color of the highlight.
ShadowColor	The color of the shadow.

Border Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

Browse for Folder

This control allows you display a browse for folder dialog, and specify whether your users can select a folder, a computer, a printer, or a file.



Browse for Folder Control

Using the Browse for Folder Control

There are four basic steps involved in using the Browse for Folder control:

1. **Insert the control into your form.**
   Use the ActiveX Control menu to place the control anywhere on your form.

2. **Set the appropriate properties.**
   Define the behavior and appearance of the dialog by setting the custom control's properties, either through the properties sheet or programmatically through VBA code. Refer to the property list below for information about customizing the dialog.

3. **Call the dialog.**
   Use the Show method to display the browse for folder dialog.

4. **Get the Return Information.**
   When the user selects a path from the dialog, the Browse for Folder control sets the Value property to the path that was selected.

Browse for Folder Properties

Name	Description
BrowseForComputer	True to limit selection to a computer, or false to allow folder/file selection as well. Note that this property only works when the EditBox property is set to False.
BrowseForPrinter	True to limit selection to a printer, or false to allow folder/file selection.
BrowseIncludeFiles	True to allow file selection, or false to limit selection to the folder level.
DontGoBelowDomain	True to disallow selecting a network computer.
EditBox	True to display a field for typing a path, or false to force the user to use the treeview.
ReturnFSAncestors	True to limit selection to file system ancestors. (An ancestor is a subfolder that is beneath the root folder.) If the user selects an ancestor of the root folder that is not part of the file system, the [OK] button is disabled.
ReturnOnlyFSDirs	True to limit selection to file system directories. If the user selects folders that are not part of the file system, the [OK] button is disabled.
Title	The string displayed in the dialog box.
Validate	True to validate the path that the user selects. If the path is invalid, the [OK] button is disabled.
Value **Default** **Databound**	The selected path.

Browse for Folder Methods

Name	Description
Show	Show the Browse for Folder Dialog.

CD Player

The CD Player control plays audio compact disks on computers that have the necessary hardware and software.

Using the CD Control

After you insert the CD Player on your form, the first step is to initialize the control using the Open method. Once opened, the other methods are available. Use the Play, Pause and Stop methods to control the CD, and use the Minute, Second, and Track properties to control which part of the CD to play. You can also use the Minutes, Seconds and Tracks properties to determine total time, elapsed time, and remaining time.

Responding to Events

The CD Player control supports two events:

- The OnNewPosition event fires every time the CD reaches a new "second". In other words, while the CD is playing, this event fires 60 times a minute. Use this property to update display controls that show the CD playing time.
- The OnNewTrack event fires every time the CD Player control reaches a new track. Use this event to update displays control that show the current track number.

Closing Down

When your application is done with a CD, it should call the Close method. This ensures that the CD Player control correctly shuts down and releases all of its resources.

CD Player Properties

Name	Description
Minute	The current minute position (this property is Read/Write).
Minutes	The number of minutes on a track.
Repeat	True to repeat the CD.
Second	The current second position.
Seconds	The number of seconds remaining on a track.
Track	The current track.
Tracks	The number of tracks on a CD.

CD Player Methods

Name	Description
Close	Closes the CD player resource.
Eject	Ejects the CD.
Open	Opens the CD player resource.
Pause	Pauses at the current play position.
Play	Plays the CD.
Stop	Stops playing the CD.

CD Player Events

Name	Description
OnNewPosition	Fires when a new position on the CD is reached.
OnNewTrack	Fires when a new track is started.

Clipboard

The Clipboard control places and retrieves text and images from the Windows Clipboard without using Windows API calls.

Using the Clipboard Control

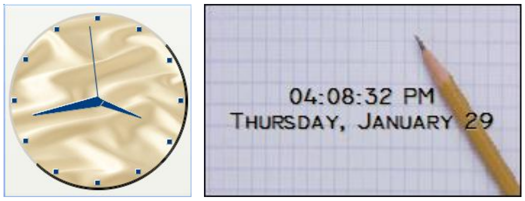
First, place the Clipboard control on your form. The Clipboard control does not have any design time properties, however it has five methods. Use the GetPicture /GetText methods to get the value of the clipboard into a variable or control, the SetPicture/SetText methods to write text to the clipboard, or the Clear method to clear the contents of the clipboard.

Clipboard Methods

Name	Description
Clear	Clears the contents of the clipboard.
GetPicture	Retrieves the image from the clipboard.
GetText	Retrieves text from the clipboard.
SetPicture	Copies an image to the clipboard.
SetText	Copies text to the clipboard.

Clock

The Clock control displays an analog or digital clock, with either 12-hour or 24-hour time format. You can optionally specify an alarm time to trigger an event when the specified time is reached.



Clock Control

Using the Clock Control

Place the Clock control on your form, set the value, and set the ClockStyle to either digital or analog. You can also set other properties, such as colors, display font, hand style, and background picture.

Setting Alarm Times

To use the Alarm feature, set the AlarmTime property. When the specified time is reached, the Clock control triggers the OnAlarm property. Write the code in this event’s Event Procedure to perform the desired action.

Clock Properties

Name	Description
AlarmEnable	True to enable the alarm.
AlarmTime	The time that the OnAlarm event should fire.
BackColor	The background color of the control.
ClipPicture	True to clip the picture to the clock’s circle. False to fit the clock control.
ClockStyle	The style of the clock.
EffectOffset	The offset of the text effect.
EffectStyle	The display style of the clock face.
Font	The font of the digital clock.
ForeColor	The color of the digital clock’s text.
HighlightColor	The highlight color.
HandStyle	The display style of the clock hands.
LongDate	True to display a long date.
Picture	The graphic displayed in the control.
ShadowColor	The shadow color.
ShowDate	True to show the date on the digital clock.
ShowHourTicks	True to show Hour tick marks on the analog clock.
ShowMinuteTicks	True to show minute tick marks on the analog clock.
ShowSeconds	True to show seconds.
TextStyle	The effect to apply to the text displayed in a Digital clock.
ThemeEnable	True to respect the appearance of the Windows Theme.
Time	The time displayed in the control.
Units	The units of the digital clock.

Clock Methods

Name	Description
LoadPicture	Loads a picture from disk into the Picture property.

Clock Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

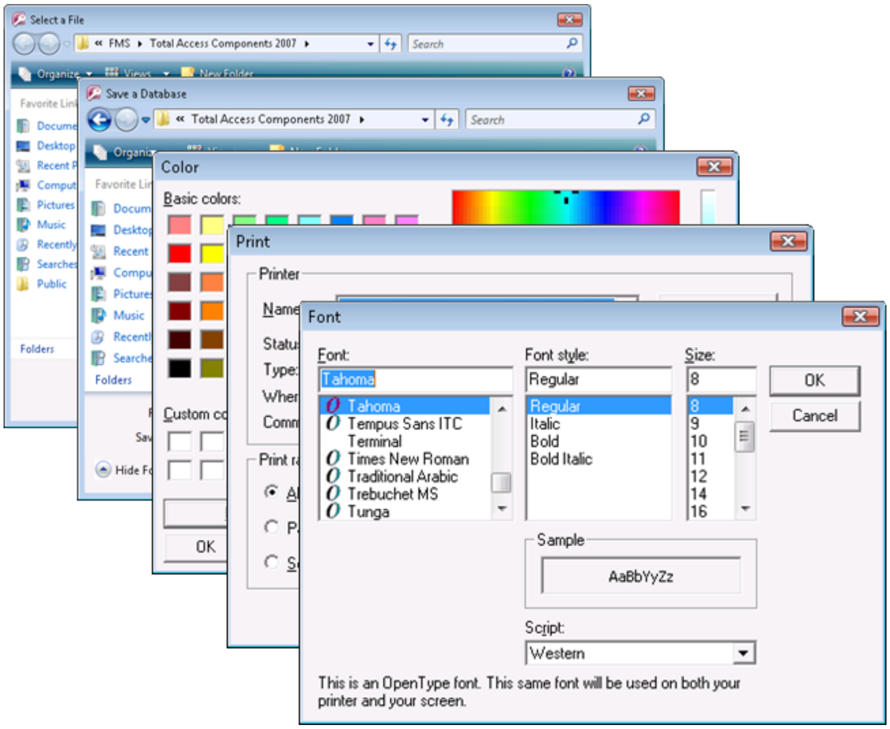
OnAlarm	Fires when AlarmTime is reached.

Common Dialog

If you have ever worked with Windows applications, you are familiar with the standard dialogs that appear when you open or save a file, or select a printer or font. These Windows dialogs are available to developers through the Windows Application Programming Interface (API). Unfortunately, using the API can be difficult for many programmers because of the complicated interface issues associated with the API.

The Common Dialog control eliminates most of these problems by encapsulating the API calls into an easy-to-use control. The following Windows dialogs are available through the Common Dialog control:

- Open Dialog (for selecting a file to open)
- Save Dialog (for selecting a name and path to save a file)
- Select Color Dialog (for color selection)
- Printer Dialog (for setting printer options)
- Font Dialog (for font selection)
- Help Dialog (opens specified HLP or CHM help file directly)



Font: Tahoma

Font style: Regular

Size: 8

Script: Western

This is an OpenType font. This same font will be used on both your printer and your screen.

Common Dialog Controls

Using the Common Dialog Control

Follow these steps to use the Common Dialog custom control:

Step 1 – Determine Which Part of the Custom Control to Use

First, determine which Windows Dialog you want to use: Open File, Save File, Select Color, Printer, Font, or Help.

Step 2 – Prepare the Dialog

Before calling the dialog, set the custom control's properties through the control's property page, or by setting the properties programmatically with VBA code. If the values for the properties are not going to change at runtime, you may find it easier to set the properties in the property sheet. If the property settings may change based on conditions that you cannot pre-define, set the properties in VBA code.

In Step 1, you determined which dialog to use. Now you must identify the properties that apply to that specific dialog. For more information on the properties that apply to each of the dialogs, see **Common Dialog Properties** on page 59.

Step 3 – Call the Dialog

Each of the Windows dialogs in the Common Dialog control has an associated Show method to display the dialog. For example, to display the Open File dialog, call the ShowOpen method.

Step 4 – Gather Return Information

As soon as your VBA code calls the dialog, the code is suspended. When you close the dialog using the [OK] or [Cancel] button, your code continues to execute. The one exception to this is the ShowHelp method, which performs the requested Help action, and immediately returns control to your VBA code.

Once the dialog has closed, you can gather information about what was done in the dialog. For example, if the Open File dialog was displayed, you probably want to retrieve the value of the path and file selected. To do this, read the values of the properties related to the dialog you are using.

Detecting the [Cancel] Button

Each common dialog has both an [OK] and a [Cancel] button. You can detect if the [Cancel] button is pressed on a dialog by setting the CancelError property of the custom control to True. When this property is set to True, pressing the [Cancel] button in the dialog triggers a VBA error, which you can trap for in your code to determine if the [Cancel] button was pressed.

Common Dialog Properties

Name	Description
CancelError	True to generate an error when the dialog box is canceled.
Color	The Font or Color Dialog color.
Copies	The number of copies to be printed.
DefaultExt	The default extension for the File Open and File Save dialogs.
DialogTitle	The string displayed in the title bar of the dialog box.
FileName	The path and file name of a selected file.
FileTitle	The file to open or save.
Filter	The filter for the File dialog.
FilterIndex	The filters that are displayed in the Type list of a dialog box.
Flags **Default**	The different options for the dialog boxes.
FontBold	True to show the font with a bold style in the Font dialog.
FontItalic	True to show the font with an italic style in the Font dialog.
FontName	The name of the font in the Font dialog.
FontSize	The size of the font in the Font dialog.
FontStrikeOut	True to show the font with strike out style in the Font dialog.

FontUnderline	True to show the font with underline style in the Font dialog.
FromPage	The From text box used by the Printer dialog.
hDC	A Windows Device Context used by Printer dialog.
HelpCommand	The type of on-line Help requested.
HelpContext	The context ID of the requested Help topic.
HelpFile	The path of the help file to display (HLP or CHM).
HelpKey	The keyword that identifies the requested Help topic.
InitDir	The initial directory/folder.
Max	The maximum font size or printer page number to show.
MaxFileSize	The maximum size of the file name opened.
Min	The minimum font size or printer page number to show.
PrinterDefault	True to set the default printer settings to the user's selections.
ToPage	The To text box used by the Print dialog.

Common Dialog Methods

Name	Description
ShowColor	Show the Color Common Dialog.
ShowFont	Show the Font Common Dialog.
ShowHelp	Show Windows help (HLP) or HTML help (CHM) file.
ShowOpen	Show the Open File Common Dialog.
ShowPrinter	Show the Printer Common Dialog.
ShowSave	Show the Save File Common Dialog.

Cursor

The cursor control is an invisible control that allows the user to specify a custom image as the form's cursor.

Using the Cursor Control

There are two basic steps involved in using the Cursor control:

Insert the control into your form.
Use the ActiveX Control menu to place the control on your form.

Set the Picture property.
Select the image to use as the form's cursor in design mode, or use the LoadPicture method to set the picture at runtime.

Tip

The Cursor control supports Bitmap, Cursor, Icon and JPEG files.

Note that the control supports Icon files (*.ico) that are 16x16, with 4bit color depth, and that use the standard Windows color palette. If the Cursor control does not properly display your Icon file, the icon may not be saved with supported size or color.

Cursor Properties

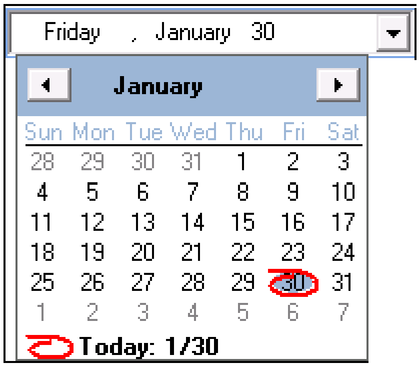
Name	Description
Enabled	True to show the custom cursor.
Picture	The graphic to display as the cursor.

Cursor Methods

Name	Description
LoadPicture	Load a picture from disk into the Picture property.

DateTime Picker

The DateTime Picker control allows users to view and select a date or time in the appropriate format, and optionally allow a drop-down calendar.



Friday , January 30

January

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Today: 1/30

DateTime Picker Control

Using the DateTime Picker Control

There are three basic steps involved in using the DateTime Picker control:

1. **Insert the control into your form.**
   Use the ActiveX Control menu to place the control on your form.
2. **Set the appropriate properties.**
   Define the date or time format, and the appearance of the control using the available properties.

3. **Set the Value**
   Set the control's value in VBA code, or bind the control to a date/time field in a table. See page 26 for details about databinding.

Customizing Date/Time Format

When you set the Format property to Custom, the DateTime Picker control uses the custom format from that you specify in the CustomFormat property. For instance, if the CustomFormat property is set to:

dddd MMM. dd, yyyy

The date is displayed as:

Saturday Aug. 8, 2009

The CustomFormat property recognizes the following values:

Value	Description	Example
d	The one- or two-digit day.	May **1** or May **10**
dd	The two-digit day.	May **01** or May **10**
ddd	The three-character weekday abbreviation.	Wed
dddd	The full weekday name.	Wednesday
h	The one- or two-digit hour in 12-hour format.	**9**:00 or **12**:00
hh	The two-digit hour in 12-hour format.	**09**:00 or **12**:00
H	The one- or two-digit hour in 24-hour format.	**9**:00 or **18**:00
HH	The two-digit hour in 24-hour format.	**09**:00 or **18**:00
m	The one- or two-digit minute.	09: **5** or 09:**15**
mm	The two-digit minute.	09:**05** or 09:**15**
M	The one- or two-digit month number.	**1**/14 or **10**/14
MM	The two-digit month number.	**01**/14 or **10**/14
MMM	The three-character month abbreviation.	Jan
MMMM	The full month name.	January
t	The one-letter AM/PM abbreviation.	1:15 **A** or 1:15 **P**
tt	The two-letter AM/PM abbreviation.	1:15 **AM** or 1:15 **PM**
yy	The last two digits of the year.	Jan. 14, **04**
yyyy	The full four-digit year.	Jan. 14, **2004**

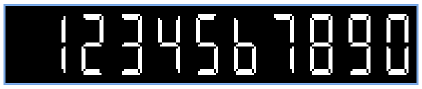
DateTime Picker Properties

Name	Description
CustomFormat	The format to use when the Format property is set to "Custom" (see the table above for details).
Format	The format to use: LongDate, ShortDate, Time, or Custom.
ShowCalendar	True to allow users to select a date from a drop-down calendar.

ShowNone	True to show a check box, which if unchecked, disallows typing in the DateTime Picker control. Note that when ShowNone is True, users can still use the Calendar drop-down to select a date. Set ShowCalendar to False to disable this.
Value **Default** **Databound**	The value of the control, for instance: 8/8/2009 5:10:46 PM This property cannot be set at design time; it can be set programmatically or bound to a Date/Time field.

Digital Display

The Digital Display control is a databound control that places digital format numbers on a form. This format is similar to the appearance of the displays on calculators and other electronic devices. A variety of formatting options are available, giving you complete control over the presentation.



Digital Display Control

Using the Digital Display Control

To use this control, simply place it on the form and set its Value property. You can set this property using the property sheet or through VBA code. You can also set the alignment to left or right. The default alignment is left, but if you want the Digital display to show characters from right to left (like most calculators), set the Alignment property to right.

Displaying Alphabetic Characters

Each digit in the Digital Display control is made up of eight segments. These segments are enabled or disabled to display the numbers. You may also use alphabetic (a-z) characters in the Value property of this control. Since all letters cannot be represented with eight segments, some letters may not appear correctly. The following letters are properly displayed:

A,B,b,C,c,D,d,E,F,g,H,h,I,i,J,L,l,N,n,O,o,P,R,r,S,U,u,Y,Z

Displaying Other Characters

The Digital Display control also displays some other useful characters. The following other characters are properly displayed: **+ - . ,**

Digital Display Properties

Name	Description
Alignment	The alignment of the text in the Digital Display.

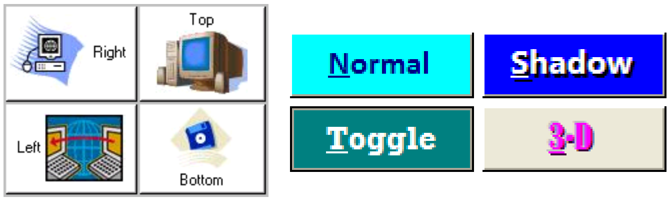
BackColor	The background color of the control.
ForeColor	The color the Digital Display.
Value **Default** **Databound**	The text to display in the Digital Display.

Digital Display Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

Enhanced Button

The Enhanced Button control displays standard and toggle buttons, with customizable pictures and captions. You can place the picture to the left, right, above, or below the caption, and specify a separate caption and picture to display when the button is “up” (not pressed in) and “down” (pressed in).



Enhanced Button Control

Using the Enhanced Button Control

The Enhanced Button is a databound control that you can use in the same way as a standard Access button. You can add code to perform your desired action in the OnClick event.

Set the control’s properties at design time using the property sheet, or in runtime using VBA code. In general, the Enhanced Button properties can be set at design time and don’t need to change at runtime. However, you may want to use VBA code to read or set property values. This code checks the Value property to determine if the button is toggled up (not pressed in) or down (pressed in):

```
If enbDemo = True Then
  MsgBox "Button is down."
Else
  MsgBox "Button is up."
End If
```

Note that in the first line, Value is implied since it’s the default property.

Setting Button Pictures

The easiest way to set the up and down picture properties is at design time using the property sheet. You can also load pictures programmatically at runtime using the LoadPicture and LoadPictureDown methods. When loading pictures at runtime, take note of these details:

- You can only assign bitmap files that are saved to disk. You cannot assign a picture from an image control or another button.
- Once you have changed the picture property in code, you cannot revert to the original picture that you set in design mode.
- The pictures you assign with the LoadPicture or LoadPictureDown methods do not permanently change the button—the change is made during run-time only.

Enhanced Button Properties

Name	Description
Appearance	The visual style of the button.
BackColor	The background color of the button.
ButtonStyle	The style of the button: standard or toggle.
Caption	The text to display on the button face.
CaptionDown	The text to display when the button is in the down state.
CursorOnHover	The appearance of the cursor when hovering on the button: Default or HyperlinkHand.
EffectOffset	The number of Pixels to offset shadowed and embossed text.
Enabled	True to allow the user to manipulate the control.
Font	The font of the text.
ForeColor	The color of the text.
HighlightColor	The color of the highlight text when TextStyle is set to Three-D.
Picture	The bitmap, metafile, or icon to display.
PictureDown	The bitmap, metafile, or icon when the button is down.
PicturePosition	The position of the bitmap relative to the text.
ShadowColor	The color to use for shadowed text.
TextStyle	The effect to apply to the text.
Value **Default** **Databound**	The toggle state of the button.

Enhanced Button Methods

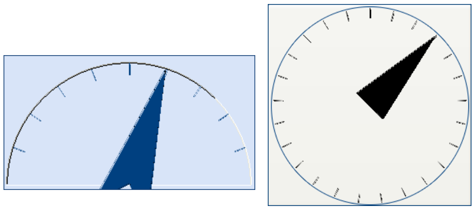
Name	Description
LoadPicture	Loads a picture into the Picture property at run time.
LoadPictureDown	Loads a picture into the PictureDown property at run time.

Enhanced Button Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

Gauge

The Gauge control is a databound control that displays quantitative information in a visual format that is easy to understand. You can select two different styles (half circle and full circle) and a variety of formatting options. Use the Gauge control to show numeric data.



Gauge Control

Using the Gauge Control

First, place the Gauge control on your form. Next, use the property sheet to determine the appearance of the gauge using the following properties: EffectStyle, GaugeStyle, LineWidth, NeedleStyle, StartLocation, TickFrequency, and TickStyle.

You can use the Value property to change the position of the needle, and optionally define minimum and maximum values using the Min and Max properties. If you try to set the Value property to a number that exceeds the Min and Max properties, a runtime error occurs.

Gauge Properties

Name	Description
BackColor	The background color of the control.
EffectStyle	The effect applied to the Dial.
EnableTheme	True to respect the appearance of the Windows Theme.
ForeColor	The color of the needle.
GaugeStyle	The display style of the gauge.
LineWidth	The width of the needle in Pixels.
Max	The maximum value.
Min	The minimum value.
NeedleStyle	The display style of the needle.
Picture	The graphic displayed in the control.
StartLocation	The starting location of the gauge dial.
TickFrequency	The frequency of the tick marks.
TickStyle	True to show Tick marks.
Value **Default** **Databound**	The value of the control.

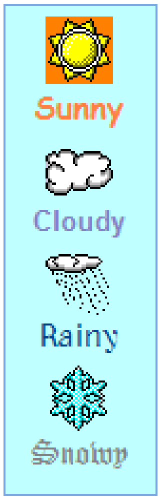
Clock Methods

Name	Description
LoadPicture	Loads a picture from disk into the Picture property.

Gauge Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

Icon Menu



Icon Menu Control

The Icon Menu control allows an alternate form of menu selection. This control displays a series of icons similar to Microsoft Outlook.

Using the Icon Menu Control

There are three basic steps for using the Icon Menu control:

Step 1 - Insert the Icon Menu

Insert the Icon Menu control into your form, and place it in the desired position.

Step 2 - Modify the Icon Menu Control's Properties

Open the Icon Menu control's property sheet by right clicking on the control, selecting FMS Icon Menu Control Object, and choosing Properties. Each icon has a caption, picture, color, and font options, which are set in the Item Properties of the property sheet. Select an individual icon to modify by changing the Item Number property. You can also set the orientation of the icons to horizontal, vertical, or flowed.

Step 3 - Write Code to Handle the Selection of Icons in the Control

Write code to respond to the OnSelect event of the Icon Menu control and perform the appropriate task.

Icon Menu Properties

Name	Description
BackColor	The background color of the button.
Enabled	True to allow the user to manipulate the control.
Font	The font for the text. This property is obsolete (replaced by ItemFont).
ForeColor	The color of the text. This property is obsolete (replaced by ItemForeColor).
ItemBackColor	The background color of the specified menu item.
ItemCount	The number of menu items.
ItemFont	The font for the text of the specified menu item.
ItemForeColor	The color of the text for the specified menu item.
ItemIcon	The icon of the specified menu item.
ItemText	The text of the specified menu item.
LargeIcons	True to show large icons.
Orientation	The orientation of the control.

Icon Menu Methods

Name	Description
LoadIcon	Loads a picture into the ItemIcon property.

Icon Menu Events

Name	Description
OnHighlight	Fires when an item is highlighted.
OnSelect	Fires when an item is selected.
OnUnHighlight	Fires when an item loses the highlight.
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

INI File

The INI File control is a non-visible control that makes it easy to read values from, and write values to Windows Initialization (INI) files. INI files are simple text files that consist of Sections and Keys. A section is identified by title text enclosed in square brackets ([ ]). All entries after the title, and up to the next title are part of that section.

16-bit versions of Windows, and some Windows applications store settings in these initialization files. In addition, your application can take advantage of INI files to read and write information that is saved between invocations.

Using the INI File Control

The INI File control only has one property and two methods, making it easy to use. First, identify the location and name of the INI file using the FileName property. Then, use the GetValue method to retrieve a value from the INI file, or the SetValue method to write text.

INI File Properties

Name	Description
FileName	The full path and filename of the INI file to use.

INI File Methods

Name	Description
GetValue	Retrieves a value.
SetValue	Sets a value.

Marquee

The Marquee control is a databound control that displays scrolling text on your forms.

Total Access Components 2

Marquee Control

Using the Marquee Control

To use this control, place it on the form and set its Value property, or bind it to a field. Optionally set the Direction, Interval, and Stepping properties to customize the direction of the scrolling text, how quickly the text scrolls, and how many pixels the text moves each time it is updated (respectively).

Marquee Properties

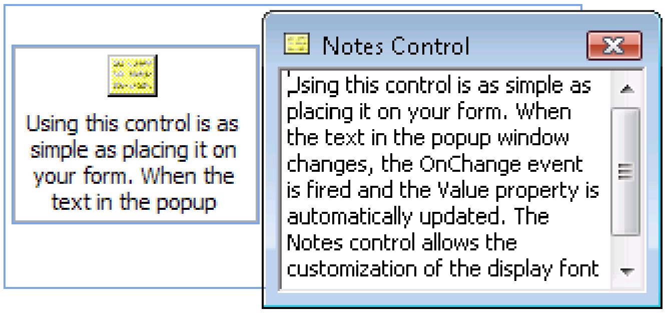
Name	Description
BackColor	The background color of the Marquee.
Direction	The direction of the scrolling text.
EffectOffset	The number of Pixels to offset shadowed and embossed text.
Font	The font for the text.
ForeColor	The color of the text.
HighlightColor	The color of the highlight text when TextStyle is set to Three-D.
Interval	The delay between steps.
ShadowColor	The color to use for shadowed text.
Stepping	The size of the steps.
TextStyle	The effect to apply to the text identified by the Value property.
ThemeEnable	True to respect the appearance of the Windows Theme.
Value **Default** **Databound**	The text to display in the control.
WordWrap	True to wrap the text if it’s wider than the control.

Marquee Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

Notes

The Notes control is a databound control that adds a popup note to your application. This control displays as an icon on your form. Double clicking it opens a window for viewing and editing the text.



Using this control is as simple as placing it on your form. When the text in the popup

Using this control is as simple as placing it on your form. When the text in the popup window changes, the OnChange event is fired and the Value property is automatically updated. The Notes control allows the customization of the display font

Notes Control

Using the Notes Control

Using this control is as simple as placing it on your form. When the text in the popup window changes, the OnChange event fires and the Value property is automatically updated. The Notes control allows the customization of the display font and colors of the control.

Notes Properties

Name	Description
BackColor	The background color of the Note.
EnableTheme	True to respect the appearance of the Windows Theme.
Font	The font for the text.
ForeColor	The color of the text.
MaxText	The maximum length of the text.

NoteBackColor	The background color of the popup window.
NoteForeColor	The foreground color of the popup window.
Opened	Whether the popup window is open.
TextLocation	The location of the text within the control.
Title	The title of the popup window.
Value **Default** **Databound**	The text contained in the control.

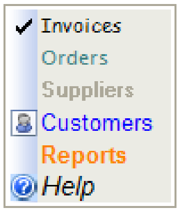
Notes Methods

Name	Description
Show	Shows the popup window.

Notes Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.
OnChange	Fires when the user changes the value of the control.

Popup Menu



The Popup Menu control displays a Windows-style context or popup menu when a user right clicks on a control or form.

Using the Popup Menu control

There are four basic steps for using the Popup Menu control:

Step 1 - Determine which controls need a popup menu

Plan the functionality and menu choices that you want to make available when the user right clicks.

Step 2 - Modify the Popup Menu control's properties

Open the Popup Menu control's property sheet by right clicking on the control, selecting FMS Popup Menu Control Object, and choosing Properties. Menu items are created by referencing each item number. Each menu choice is displayed using its ItemText property, and can be checked, enabled/disabled, or can act as a separator. Each item can be customized with an icon, and with color and font options.

Step 3 - Write code to display the popup menu

The typical way to handle this is to trap the clicking of the right mouse button. In this case, the code (to call the Popup Menu control's Show method) is put in the form or control's MouseDown or MouseUp event and is run if the mouse button pressed is the right button (`Button = 2`).

Step 4 - Write code to handle the menu choice selected

In the Popup Menu control's OnSelect event, set up a way to handle each menu choice, usually by using a select case block.

Popup Menu Properties

Name	Description
EnableTheme	True to respect the appearance of the Windows Theme.
ItemBackColor	The background color of the specified menu item.
ItemCount	The number of menu items.
ItemChecked	The checked property of the specified item.
ItemEnabled	The enabled property of the specified item.
ItemFont	The font for the text of the specified menu item.
ItemForeColor	The color of the text for the specified menu item.
ItemIcon	The icon of the specified menu item.
ItemSeparator	The separator property of the specified item.
ItemText	The text of the specified menu item.

Popup Menu Methods

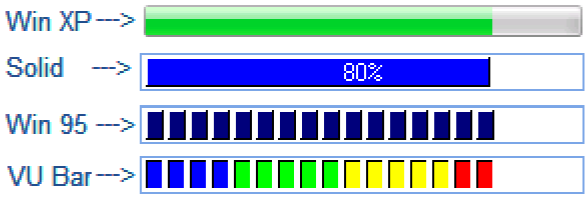
Name	Description
LoadIcon	Loads a picture into the ItemIcon property.
Show	Displays the popup menu.

Popup Menu Events

Name	Description
OnSelect	Fires when a menu item is selected.

Progress Meter

The Progress Meter is a databound control that is designed to provide visual feedback to your users about various processes, such as the duration of an event or the magnitude of a value. Your user can make visual estimates of the time remaining to complete an event, or the percent of a numeric value without having to translate numbers or time units mentally.



Progress Meter Control

The FMS Progress Meter has a wide variety of styles. The bar style of the meter can be either solid or segmented. If the meter is segmented, the segments may either be flat, raised, or sunken. A segmented meter may also use different colors for different segments. The Visual Unit (VU) style mimics the look of the VU meter of a sound system. The meter has a caption to display the numeric value of the meter, or the percent value. The meter may be oriented either vertically or horizontally, and may be filled either from the left/bottom or the right/top.

The best way to determine which style is right for your application is to experiment with the various settings using the property sheet for the control while it is in design mode. All properties of the progress meter can also be set programmatically during run-time.

Using the Min and Max Properties

Determine the range of values that your control displays by setting the Min and Max properties. If Min is set to zero and Max is set to 100, then when the meter is filled halfway, the value of the control is 50. The Min and Max values can be set to any numeric integer value, as long as the Min value is less than the Max value. This means that if you are trying to reset the Min and Max values such that the new Min value is *higher* than the current Max value, adjust the Max value first, then change the Min value.

Using the Value Property

The Value property of the control sets the percentage of the bar that is filled in. Set the Value property to a number between the Min and Max values, and the progress meter automatically displays the appropriate bar or number of segments.

Using the Visual Unit (VU) Color Properties

The visual display of the VU-style progress meter can be further enhanced by specifying the colors to be used for up to four sets of segments. Specify the colors you wish to use for the segment sets by selecting the color for the VUColor1, VUColor2, VUColor3 and VUColor4 properties. Then specify the percentage point at which the segments change from one color to another by setting the corresponding VUPercent2, VUPercent3 and VUPercent4 properties. There is no VUPercent1 property since the VUColor1 property is used for the first segment set.

Tip

When the BarStyle is set to Windows XP, the control's colors are determined by the Windows Display properties, rather than by the control's color properties. This allows the control to conform to the system colors or theme specified by the user.

Displaying Text with the Caption Property

The Caption property of the meter can be used to display text in the Progress Meter. You may choose to display the actual numeric value reflected by the meter or the percentage. The Progress Meter recognizes the special case of using a percent sign (%) in the caption. The % is automatically replaced with the actual percentage value. For example, if the caption is set to "%", the actual caption displays "55 %" (where 55 is the actual percentage value).

Progress Meter Properties

Name	Description
BackColor	The background color for the "undone" portion of the progress meter.
BarColor	The color for the "done" or bar portion of the progress meter.
BarStyle	The style of the progress meter.
EffectStyle	The appearance of the border surrounding the control.
Caption	The string to place in front of the Value property.
FillStyle	Where the bar begins to fill.
Font	The font for the text.
Max	The maximum value.
Min	The minimum value.
Orientation	The orientation of the progress meter.
Value **Default** **Databound**	The value of the control.
VUColor1	The color for the first visual unit (VU) Segment.
VUColor2	The color for the second VU Segment.
VUColor3	The color for the third VU Segment.
VUColor4	The color for the fourth VU Segment.
VUPercent2	The percentage where VU Color2 starts drawing.

VUPercent3	The percentage where VU Color3 starts drawing.
VUPercent4	The percentage where VU Color4 starts drawing.

Progress Meter Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

Registry

The Windows registry is a structured database used to store setting and configuration information for Windows and for applications that run under Windows. You can see the registry by running REGEDIT.EXE. The FMS Registry control works with registry values by reading and writing data.

To use the Registry control, you need a basic understanding of the Windows Registry. The registry is organized in various top-level (or root) keys. Each of these root keys contains sub-keys, which can themselves contain sub-keys and values. Each value in a key has data associated with it. A value is identified by three pieces of information: the root key, the key name, and the value name.

Using the Registry Control

There are no properties for the Registry control, only methods. Use the control’s methods to read, write, edit, and remove registry keys and values, and to register applications.

Using the Register Method

The Register method is provided for applications that need to register OLE servers. You can register an OLE server by providing the file name of the server. The control automatically updates the registry with the appropriate entries.

Registry Methods

Name	Description
DeleteKey	Deletes the specified key.
DeleteKeyValue	Deletes a key value.
GetSubkeyCount	Returns the number of child keys of the specified key.
GetSubkeyString	Returns a given sub-key string.
GetKeyValue	Retrieves the data associated with the key value.
GetValueCount	Returns the number of values for the specified key.
GetValueString	Returns a given value string.
Register	Registers the specified library.
SetKeyValue	Sets the data associated with a key value.

Registry Constants

The following root keys are available:

Key Name	Value	Constant
HKEY_CLASSES_ROOT	2147483648	TacmReghKeyClassesRoot
HKEY_CURRENT_CONFIG	2147483653	TacmReghKeyCurrentConfig
HKEY_CURRENT_USER	2147483649	TacmReghKeyCurrentUser
HKEY_DYN_DATA	2147483654	TacmReghKeyDynData
HKEY_LOCAL_MACHINE	2147483650	TacmReghKeyLocalMachine
HKEY_PERFORMANCE_DATA	2147483652	TacmReghKeyPerformanceData
HKEY_USERS	2147483651	TacmReghKeyUsers

Resize

The Resize control is a non-visual control that automatically resizes controls on a form when the form is resized. The controls are resized based on their relative sizes when the Init method is called. The ResizeFonts property gives you the option to resize fonts along with the form as well.

Using the Resize Control

Using the Resize control involves three steps:

Step 1 – Initialize the Resize Control

In your forms Load event, call the Init method of the Resize control. This initializes the control and lets it determine the sizing ratios for the controls on the form.

If your form includes a header and/or footer, call the SetFormHeader and SetFormFooter methods to initialize header and footer resizing.

Step 2 – Tell the Resize Control to Resize the Controls on the Form

In your form's Resize event, call the Resize method of the Resize control. This resizes all the controls on the form.

Step 3 – Close the Resize Control

In your form's Close event, call the Close method of the Resize control. This releases any resources the control is using.

Resize Properties

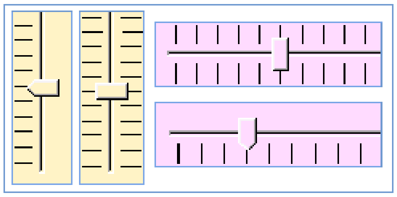
Name	Description
MaxHeight	The maximum height to resize the form.
MaxWidth	The maximum width to resize the form.
MinHeight	The minimum height to resize the form.
MinWidth	The minimum width to resize the form.
ResizeFonts	True to resize fonts with the form.

Resize Methods

Name	Description
Close	Close the control and any resources it may be using.
Init	Initialize the sizing information for the form.
Resize	Resize the form and associated controls.
SetFormFooter	Initialize the form footer sizing information.
SetFormHeader	Initialize the form header sizing information.

Slider

The Slider control is a databound control that is used to adjust numeric values by dragging or sliding the control's handle. It is commonly used in Windows Applications to adjust a continuous numeric value smoothly. The Slider control can be associated with another control to adjust that control's value, or it can be used by itself as a vehicle for input.



Slider Control

Using the Slider control

There are three basic steps for using the Slider control:

Step 1 - Determine the control to associate with the Slider control

If you want the slider to adjust the value in another control, determine which control on the form to use. Otherwise, the slider can be used by itself to update a field or a value used in a subroutine.

Step 2 - Modify the Slider control's properties

Use the Slider control's property sheet to configure the control's initial property values. The Min and Max properties should be set to the lower and upper limits of the input required. The control's Value property is used to set its initial value. The LargeChange and SmallChange properties can be used to increment a value by the specified amount when the LargeChange or SmallChange event fires.

Step 3 - Write code to handle the change in the Slider control's value

In the Slider control's OnChange event, write code to handle the new value returned by the Slider control. For example, you may want to use this event to change another control's value.

Slider Properties

Name	Description
BackColor	The color of the control's background.
Direction	The direction from which the slider increases.
Enabled	True to allow the user to manipulate the control.
LargeChange	The value to change the slider for a large change event.
Max	The maximum value (an integer).
Min	The minimum value (an integer).
Orientation	The control's orientation.
SmallChange	The value to change the slider for a small change event.
TickFrequency	The frequency of the tick marks.
TickStyle	The location of the tick marks.
Value **Default** **Databound**	The value of the control.

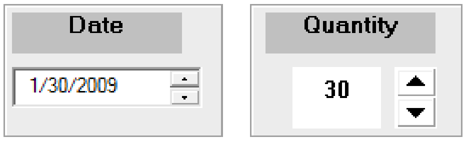
Slider Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.

KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.
OnChange	Fires when the user changes the value of the control.

Spin Button

The Spin Button is a databound control that is used to increment or decrement a numeric value or date by clicking on the control's "up" or "down" button. It is commonly used in Windows applications in conjunction with a text box containing a numeric value.



Spin Button Control

Using the Spin Button control

There are three basic steps for using the Spin Button control:

Step 1 - Determine the control to be associated with the spin control

This is usually a text box, but it can be any control that displays a numeric or date value.

Step 2 - Modify the spin control's properties

Use the Spin Button's property sheet to configure the control's initial property values. The Min and Max properties should be set to the lower and upper limits of the input required. The LargeChange and SmallChange properties can be used to increment a value by the specified amount when the OnSpin fires. To allow the spin control's OnSpin event to fire repeatedly while depressing either of its buttons, check property. The RollOver property allows the spin control's value to cycle (reset to the Min or Max value if you try to "go past" them).

Step 3 - Write code to handle the change in the spin control's value

In the spin control's OnSpin event, write code to change the associated control's value by substituting the spin control's Value property, or by adding/subtracting the spin control’s LargeChange or SmallChange property values to/from the value in the associated control.

Spin Button Properties

Name	Description
AutoIncrement	True to update the Value when the user changes the control.
ArrowStyle	The display style of the arrow.
BackColor	The background color of the control.
Enabled	Whether the user can modify the control.
ForeColor	The foreground color of the control.
HighlightColor	The highlight color of the arrow.
LargeChange	The value the Spin Button changes for a large change event.
Max	The maximum value.
Min	The minimum value.
Orientation	The control’s orientation.
RollOver	True to reset to the minimum when it exceeds the maximum, or to the maximum when it goes below the minimum.
ShadowColor	The color of the shadow.
SmallChange	The value the Spin Button changes for a small change event.
Value **Default** **Databound**	The value of the control.

Spin Button Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.
OnChange	Fires when the user changes the value of the control.
OnSpin	Fires when the user clicks the up or down button.

Splitter

Splitter Control

The splitter control provides automatic resizing of controls on your form. If the Orientation property is set to horizontal, the control resizes any controls that lie to the top or bottom. If the Orientation property is set to vertical, the control resizes any controls that lie to the left or right.

Using the Splitter Control

First, place the splitter control on your form and set the Orientation property to either horizontal or vertical. This controls the direction in which you can drag the splitter. Next, position the splitter control between the other form controls you wish to resize.

Splitter Properties

Name	Description
BackColor	The background color of the splitter.
Max	The maximum value of the splitter.
Min	The minimum value of the splitter.
Orientation	The orientation of the splitter.

Splitter Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.
OnSplit	Fires after the splitter moved.

System Information

The System Information Control is a non-visible control that retrieves computer and network information. Place a control on a form and use its properties and methods to get information without using complicated Windows API calls, and information that is not available through the Windows API.

Using the System Information Control

The System Information control doesn't have any design time properties. All of its properties are read-only. Use the properties and methods to retrieve the information you need.

Identifying Drive Letters

Several of this control's methods require the DriveLetter parameter. This parameter identifies the drive for which you want information. This number is 0-based, meaning that DriveLetter 0 is the A drive, DriveLetter 1 is B, DriveLetter 2 is C, DriveLetter3 is D, etc.

You can use these methods to determine the drives on your system. You should enumerate through all 26 numbers and check the return value. If you refer to an invalid drive number, the method triggers a VBA runtime error. Trap the error to identify an invalid drive.

System Information Properties

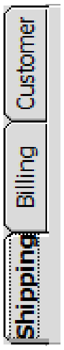
Name	Description
CapsLock	The status of the Caps Lock key.
ComputerName	The network computer name.
CPUType	The CPU type.
FreeMem	The total free memory, in bytes.
FreeMemEx	The total free memory in bytes, with support for larger values than FreeMem.
MathCo	The presence of a math coprocessor.
NumLock	The status of the Num Lock key.
SystemDir	The path and name of the Windows System directory.
TempDir	The path and name of the temporary directory.
TempFileName	The path and name of a temporary file.
UserName	The network user name.
WindowsDir	The path and name of the Windows directory.
WindowsType	The Windows type.
WindowsVersion	The Windows version.

System Information Methods

Name	Description
GetDiskType	Returns the type of the specified drive.
GetDiskVolume	Returns the volume of the specified drive.
GetFileExt	Returns the file extension.
GetFileName	Returns the filename portion of the path.
GetFilePath	Returns the path without the filename or extension.
GetFileVersion	Returns the version of the specified file.

GetFreeDiskSpace	Returns the free disk space on the specified drive. This method is obsolete; use GetFreeDiskSpaceEx instead.
GetFreeDiskSpaceEx	Returns the free disk space on the specified drive, but supports larger values than GetFreeDiskSpace.

Tab



The Tab control, also known as a tabbed divider, is one of the more useful Windows controls because it allows many other controls to be grouped or organized using a visual metaphor with which most users are familiar. The FMS Tab control achieves the same effect. This control can be databound.

Problems with Containership

If you have used Tab controls in other development environments such as Visual Basic 6, you are familiar with the concept of *containership*. For example, in a Visual Basic Tab control, each page of the tab control can *contain* other controls—when you click on different tabs, the tab control automatically displays the controls that the page contains. Unfortunately, Access does not support containership for custom controls (see page 20 for details). Because of this, the Total Access Components Tab control cannot automatically show or hide controls depending on which page is active. Fortunately, writing the code to make this work is not difficult.

The sample database contains several examples of working around this issue. When a different tab on the Tab control is selected, you can:

- Cycle through the controls and set the Visible property to True or False, depending on whether that control belongs to the current page.
- Use subform controls to group and contain objects.

Tip

The built-in Access Tab control provides a more convenient method for containership than the FMS Tab control, however the FMS Tab control provides some features not found in the Access Tab control, including databinding, Windows 3.1 appearance, and the option to displays tabs on the left, bottom, or right.

If you need any of these features, consider using the FMS Tab control. Otherwise, using the Access Tab control is generally a better solution.

Problems with Paint Over

An additional Access-related problem is that custom controls are higher in the Windows Z-Order than most other Access controls (see page 21 for details). This means if you place an Access text box control on top of a custom control in design view, and then view the form, the text box goes behind the custom control and is not visible. This happens even when you use the Bring To Front command on the Access menu. There are two ways to get around this problem:

1. Use the FMS Tab control as a “strip control” by sizing it to be tall enough to contain only the tabs. Then set the BodyStyle property of the Tab control to “Open” to display a partially complete tab control on your form. Add line up an Access rectangle control below the tab strip, and set its Special Effect property to “Raised” to complete the overall look. Switch to form view mode to ensure that all controls line up.

2. Place a Tab control on your form, size it to be a “strip control,” and set the properties to provide the desired appearance. Then create another Tab control with the same property settings, and use the “Convert To Picture” Access operation to convert the second tab control to a picture (see page 33 for details). Since the second Tab control is now a picture, it is always displayed behind other controls.

Fonts and Tab Location

If you set the TabLocation property to Left or Right, the Tab control automatically rotates the tab text. However, in Windows, only TrueType fonts can be rotated. If you use the Left or Right orientations, you must specify a TrueType font for the tab text, or the text does not display correctly. TrueType fonts are displayed in the Font list with the TT icon.

Using the Tab Control

There are three basic steps for using the tab control:

Step 1 - Set up the tab and associated controls

Place the following on the form: the tab control, a raised rectangle, and all the controls to be “contained” by each tab. Align the edges of the tab control and the rectangle (which acts as the visual container) next to each other.

Step 2 - Modify the tab control’s properties

Open the Tab control’s custom property sheet. To emulate the look of a tab control in the Windows environment, set the BodyStyle property to `0 – Open`. Set the Appearance property to `0 – Win31`, or `1 – Win95`. Set the location of the tabs (top, bottom, left, or right) with the TabLocation property, and set each tab’s caption with the TabText property. Modify an individual tab by changing the Value property.

Step 3 - Write code to handle the display of each tab on the control

By responding to the OnChange event of the tab control, you can set the Visible property for each control "contained" in the tab control, which simulates changing tabs.

Tab Properties

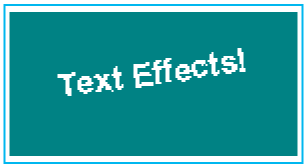
Name	Description
Appearance	The tab style.
BackColor	The background color of the control.
BodyStyle	The tab control body style.
Enabled	True to allow the user to manipulate the control.
Font	The font displayed in the tabs.
TabCount	The number of tabs in the control.
TabLocation	The location of the tabs.
TabText	The text displayed in the tab.
Value **Default** **Databound**	The currently selected tab.

Tab Events

Name	Description
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.
OnChange	Fires when the user changes the value of the control.

Text Effects

The Text Effects control is a databound control that is used to spice up the display of text on your forms and reports. You can change the angle that text is displayed, and choose various three-D styles.



Text Effects Control

Rotating Text

First, set the Value property to the text that you want to display. Then, set the other properties for the control.

To rotate text, set the Angle property to any value between 0 and 360. A value of 90 creates vertical text with the top of the text facing to the left, and 270 shows text with the top of the text on the right. A value of 180 displays the text upside down.

Set other properties, such as TextLocation, TextStyle, Font, and colors to customize the appearance of the control.

Tip

> To use rotated text, you must specify a TrueType font. Non-TrueType fonts cannot be rotated. TrueType fonts are shown in the Font list with the TT icon.

Refreshing the Display

The DoPaint method forces an immediate display of the control when you change one or more of its property values. Ordinarily, Windows automatically displays the changes as soon as it is given enough processing time. However, if you are rapidly changing the value of one or more properties (for example in a tight programming loop or in response to user input with a Slider control), the DoPaint method ensures that each change is immediately displayed.

Text Effects Properties

Name	Description
Angle	The degrees of rotation for the text.
BackColor	The background color of the control.
EffectOffset	The offset of the text effect.
Font	The font object for setting font properties for the text.
ForeColor	The color of the text.
HighlightColor	The color of the highlighted text.
ShadowColor	The color of the shadowed text.
TextLocation	The location of the text within the control.
TextStyle	The effect to apply to the text.

Value **Default** **Databound**	The text displayed in the control.
WordWrap	True to wrap the text if it’s wider than the control.

Text Effects Methods

Name	**Description**
DoPaint	Forces the control to update.

Text Effects Events

Name	**Description**
Click	Fires when the mouse is clicked over the control.
DblClick	Fires when the mouse is double clicked over the control.
KeyDown	Fires when a key is depressed.
KeyPress	Fires when a key is pressed.
KeyUp	Fires when a key is released.
MouseDown	Fires when a mouse button is depressed over a control.
MouseMove	Fires when the mouse moves over the control.
MouseUp	Fires when a mouse button is released over a control.

Timer

The timer control is a non-visual control that allows you to add multiple timers to your form. The built-in Timer control in Microsoft Access is limited to one timer per form.

Using the Timer Control

The timer control has only one property and one event. Using this control is as simple as setting the Interval property to the desired millisecond value and writing code to handle the OnTimer event.

Setting the Interval property to zero stops the timer.

Timer Properties

Name	**Description**
Interval **Default**	The number of milliseconds between OnTimer events.

Timer Events

Name	**Description**
OnTimer	Fires when the time specified by the interval has elapsed.

Wave

The Wave control plays Windows digitized audio *.WAV files and MIDI sequences through your computer's sound card. You can play, pause, stop, and rewind these audio files, and set repeat play.

Tip

For this control to work, your machine must already have the correct hardware devices and software drivers.

Using the Wave Control

Place a Total Access Components Wave control on your form. After placing and naming the control, use the property sheet to specify the name and path of the WAV or MIDI file, or write the appropriate VBA code to set the FileName property. Note that you do not have to identify the file's type as WAV or MIDI; the control does this automatically. Use the Play, Rewind, Pause, or Stop methods to control the file.

Stopping and Rewinding

The Stop method stops playing the file and rewinds to the beginning. If the Play method is called and the file plays until completion, the control also rewinds to the beginning.

Wave Properties

Name	Description
FileName	The path and name of the WAV or MIDI file.
Repeat	True to replay the file.
RepeatCount	How many times to repeat play (integer).

Wave Methods

Name	Description
Pause	Pauses the sound file.
Play	Plays the sound file.
Rewind	Rewinds the sound file to the beginning.
Stop	Stops and rewinds the sound file.

Wave Events

Name	Description
OnReplay	Fires when play starts over.

Chapter 6: Naming Conventions, Errors, and Constants

This chapter serves as a reference for the constants and errors defined by Total Access Components, and lists suggested naming conventions for Total Access Components controls.

Topics in this Chapter

- **Naming Conventions**
- **Error Handling**
- **Constants**

Naming Conventions

Using naming conventions is an important programming practice that can make development and maintenance easier. The use of consistent naming conventions is especially important with custom controls, since Access can't distinguish between different types of ActiveX controls.

This list shows the naming conventions that we use in the demo, help file, and documentation. Keep in mind that this list represents our conventions, and you aren't required to use them.

Control	Prefix
About Box	abt
AVI Player	avi
Bitmap Effects	bmp
Border	bdr
Browse for Folder	bff
CD Player	cdp
Clipboard	clp
Clock	clk
Common Dialog	cdl
Cursor	cur
DateTime Picker	dtp
Digital Display	dds
Enhanced Button	enb
Gauge	gau
Icon Menu	icm

Control	Prefix
INI Control	ini
Marquee	mrq
Notes	nte
Popup Menu	pop
Progress Meter	prg
Registry	reg
Resize	rsz
Slider	sld
Spin Button	spn
Splitter	spl
System Information	sys
Tab	tab
Text Effects	tef
Timer	tim
Wave	wav

Error Handling

When a Total Access Components control encounters an error, it returns an error code to Access, causing Access to display an error message. When programming with VBA, you can trap for specific errors using the error codes returned, and have your application act accordingly.

Example Error Checking

In the following example, the AnimationEffect property of a Bitmap Effects control is set to 9. Since this is an invalid property value, Total Access Components returns the error code *TacmErrInvalidPropertyValue*, and

Access generates an error. Trap for the specific error in code and have your application handle it appropriately.

```
Sub GenErrProc()
  ' Trap for errors
  On Error Resume Next

  ' Assign Invalid Value to Animation Effect
  Me.bmpDemo.AnimationEffect = 9

  ' If there was an error
  If Err <> 0 Then
    ' If the error was generated by the control
    If Err = TacmErrInvalidPropertyValue Then
      MsgBox "Invalid Property Value"
    Else
      ' Else unknown error
      MsgBox Error(Err)
    End If
  End If

  ' Resume standard error checking
  On Error GoTo 0
End Sub
```

Specific Errors

The following tables list all errors specific to Total Access Components and the controls in which they occur.

AVI Control

Property	Error	Constant	Cause
RepeatCount	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero

Bitmap Effects Control

Property	Error	Constant	Cause
AnimationEffect	Invalid Value	TacmErrInvalidPropertyValue	AnimationEffect is Invalid
CurrentPicture	Invalid Value	TacmErrInvalidPropertyValue	CurrentPicture is Invalid
CurrentStep	Invalid Value	TacmErrInvalidPropertyValue	Value is not between zero and Steps
Delay	Invalid Value	TacmErrInvalidPropertyValue	Value is Less than zero
StartPicture	Invalid Value	TacmErrInvalidPropertyValue	StartPicture is invalid
Steps	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero

Border Control

Property	Error	Constant	Cause
EffectStyle	Invalid Value	TacmErrInvalidPropertyValue	EffectStyle is invalid
CurveHeight	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
CurveWidth	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
BevelWidth	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero

Clock Control

Property	Error	Constant	Cause
AlarmTime	Invalid Value	TacmErrInvalidPropertyValue	AlarmTime is invalid
EffectStyle	Invalid Value	TacmErrInvalidPropertyValue	EffectStyle is invalid
ClockStyle	Invalid Value	TacmErrInvalidPropertyValue	ClockStyle is invalid
TextStyle	Invalid Value	TacmErrInvalidPropertyValue	TextStyle is invalid
Units	Invalid Value	TacmErrInvalidPropertyValue	Units is invalid

DateTime Picker Control

Property	Error	Constant	Cause
Value	Invalid Type	TacmErrInvalidPropertyValue	Value is invalid

Common Dialog Control

Method	Error	Constant	Cause
ShowColor	Dialog Cancelled	TacmErrCancelError	Dialog was cancelled
ShowFont	Dialog Cancelled	TacmErrCancelError	Dialog was cancelled
ShowHelp	File Not Found	TacmErrFileNotFound	Help file is missing
ShowOpen	Dialog Cancelled	TacmErrCancelError	Dialog was cancelled
ShowPrinter	Dialog Cancelled	TacmErrCancelError	Dialog was cancelled
ShowSave	Dialog Cancelled	TacmErrCancelError	Dialog was cancelled

Enhanced Button Control

Property	Error	Constant	Cause
Appearance	Invalid Value	TacmErrInvalidPropertyValue	Appearance is invalid
ButtonStyle	Invalid Value	TacmErrInvalidPropertyValue	ButtonStyle is invalid
EffectOffset	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
PicturePosition	Invalid Value	TacmErrInvalidPropertyValue	PicturePosition is invalid
TextStyle	Invalid Value	TacmErrInvalidPropertyValue	TextStyle is invalid

Gauge Control

Property	Error	Constant	Cause
EffectStyle	Invalid Value	TacmErrInvalidPropertyValue	EffectStyle is invalid
GaugeStyle	Invalid Value	TacmErrInvalidPropertyValue	GaugeStyle is invalid
LineWidth	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
Max	Invalid Value	TacmErrInvalidPropertyValue	Value is less than Min
Min	Invalid Value	TacmErrInvalidPropertyValue	Value is greater than Max
NeedleStyle	Invalid Value	TacmErrInvalidPropertyValue	NeedleStyle is invalid
TickFrequency	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
TickStyle	Invalid Value	TacmErrInvalidPropertyValue	TickStyle is invalid
Value	Invalid Value	TacmErrInvalidPropertyValue	Value is not between Min and Max

Icon Menu Control

Property	Error	Constant	Cause
ItemBackColor	Invalid Index Value	TacmErrInvalidIndexValue	Index is invalid
ItemCount	Invalid Value	TacmErrInvalidPropertyValue	ItemCount is invalid
ItemFont	Invalid Value	TacmErrInvalidIndexValue	Index is invalid
ItemForeColor	Invalid Index Value	TacmErrInvalidIndexValue	Index is invalid
ItemIcon	Invalid Index Value	TacmErrInvalidIndexValue	Index is invalid
ItemText	Invalid Index Value	TacmErrInvalidIndexValue	Index is invalid
Orientation	Invalid Value	TacmErrInvalidPropertyValue	Orientation is invalid

Method	Error	Constant	Cause
LoadIcon	Invalid Index Value	TacmErrInvalidIndexValue	Index is invalid

INI File Control

Method	Error	Constant	Cause
SetValue	Path or File Access Error	TacmErrPathFileAccess	Value cannot be written to initialization file
SetValue	Invalid Use of Null	TacmErrNull	Null value is passed to method
SetValue	Invalid Value Type	TacmErrNotSupported	Variant type not supported

Marquee Control

Property	Error	Constant	Cause
EffectOffset	Invalid Value	TacmErrInvalidPropertyValue	EffectOffset is invalid
TextStyle	Invalid Value	TacmErrInvalidPropertyValue	TextStyle is invalid
Direction	Invalid Value	TacmErrInvalidPropertyValue	Direction is invalid
Stepping	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
Interval	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero

Popup Menu Control

Property	Error	Constant	Cause
ItemChecked	Invalid Index Value	TacmErrInvalidIndexValue	Index is less than zero or greater than ItemCount-1
ItemCount	Invalid Value	TacmErrInvalidIndexValue	Value is zero or less
ItemEnabled	Invalid Index Value	TacmErrInvalidIndexValue	Index is less than zero or greater than ItemCount-1
ItemSeparator	Invalid Index Value	TacmErrInvalidIndexValue	Index is less than zero or greater than ItemCount-1
ItemText	Invalid Index Value	TacmErrInvalidIndexValue	Index is l than zero or greater than ItemCount-1

Progress Meter Control

Property	Error	Constant	Cause
BarStyle	Invalid Value	TacmErrInvalidPropertyValue	BarStyle is invalid
EffectStyle	Invalid Value	TacmErrInvalidPropertyValue	EffectStyle is invalid
FillStyle	Invalid Value	TacmErrInvalidPropertyValue	FillStyle is invalid
Max	Invalid Value	TacmErrInvalidPropertyValue	Value is less than Min
Min	Invalid Value	TacmErrInvalidPropertyValue	Value is greater than Max
Orientation	Invalid Value	TacmErrInvalidPropertyValue	Orientation is invalid
Value	Invalid Value	TacmErrInvalidPropertyValue	Value is not between Min and Max
VUpercent2	Invalid Value	TacmErrInvalidPropertyValue	Value is not between zero and 100
VUpercent3	Invalid Value	TacmErrInvalidPropertyValue	Value is not between zero and 100
VUpercent4	Invalid Value	TacmErrInvalidPropertyValue	Value is not between zero and 100

Registry Control

Method	Error	Constant	Cause
GetKeyValue	Path Not Found	TacmErrPathNotFound	Key is not in registry
GetSubKeyCount	Path Not Found	TacmErrPathNotFound	Key is not in registry
GetSubKeyString	Path Not Found	TacmErrPathNotFound	Key is not in registry
GetSubKeyString	Invalid Index Value	TacmErrInvalidIndexValue	Value is not between zero and GetSubKeyCount
GetValueCount	Path Not Found	TacmErrPathNotFound	Key is not in registry
GetValueString	Path Not Found	TacmErrPathNotFound	Value is not between zero and GetValueCount
Register	Invalid Filename	TacmErrPathFileAccess	File is not found
Register	Can’t Register Server	TacmErrCantRegister	Could not register server
SetKeyValue	Can’t Write to Registry	TacmErrPermissionDenied	Could not write to registry
SetKeyValue	Invalid Use of Null	TacmErrNull	Null value passed to method
SetKeyValue	Invalid Value Type	TacmErrNotSupported	Variant type is unsupported

Slider Control

Property	Error	Constant	Cause
Direction	Invalid Value	TacmErrInvalidPropertyValue	Direction is invalid
LargeChange	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
Max	Invalid Value	TacmErrInvalidPropertyValue	Value is less than Min
Min	Invalid Value	TacmErrInvalidPropertyValue	Value is greater than Max
Orientation	Invalid Value	TacmErrInvalidPropertyValue	Orientation is invalid
SmallChange	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
TickFrequency	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
TickStyle	Invalid Value	TacmErrInvalidPropertyValue	TickStyle is invalid
Value	Invalid Value	TacmErrInvalidPropertyValue	Value is not between Min and Max

Spin Button Control

Property	Error	Constant	Cause
ArrowStyle	Invalid Value	TacmErrInvalidPropertyValue	ArrowStyle is invalid
LargeChange	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
Max	Invalid Value	TacmErrInvalidPropertyValue	Value is less than Min
Min	Invalid Value	TacmErrInvalidPropertyValue	Value is greater than Max
Orientation	Invalid Value	TacmErrInvalidPropertyValue	Orientation is invalid
SmallChange	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
Value	Invalid Value	TacmErrInvalidPropertyValue	Value is not between Min and Max

System Information Control

Method	Error	Constant	Cause
GetDiskVolume	Invalid Index Value	TacmErrInvalidIndexValue	Disk drive index is invalid
GetFreeDiskSpace	Invalid Index Value	TacmErrInvalidIndexValue	Disk drive index is invalid
GetFreeDiskSpaceEx	Invalid Index Value	TacmErrInvalidIndexValue	Disk drive index is invalid

Tab Control

Property	Error	Constant	Cause
Appearance	Invalid Value	TacmErrInvalidPropertyValue	Appearance is invalid
BodyStyle	Invalid Value	TacmErrInvalidPropertyValue	BodyStyle is invalid
TabCount	Invalid Value	TacmErrInvalidPropertyValue	Value is zero or less
TabLocation	Invalid Value	TacmErrInvalidPropertyValue	TabLocation is invalid
TabText	Invalid Index Value	TacmErrInvalidPropertyValue	Value is not between zero and TabCount – 1
Value	Invalid Value	TacmErrInvalidPropertyValue	Value is not between zero and TabCount – 1

Text Effects Control

Property	Error	Constant	Cause
Angle	Invalid Value	TacmErrInvalidPropertyValue	Value is not between –360 and 360
EffectOffset	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero
TextStyle	Invalid Value	TacmErrInvalidPropertyValue	TextStyle is invalid

Wave Control

Property	Error	Constant	Cause
RepeatCount	Invalid Value	TacmErrInvalidPropertyValue	Value is less than zero

Constants

The Total Access Components demo files, help file, and documentation refer to specific values for properties and methods, which are represented by a set of VBA constants. These constants are available in VBA when a reference is set to the Total Access Components library. This occurs automatically when you insert a Total Access Components control into your database.

If your programming style includes the use of constants, you should use these Total Access Components constants instead of hard-coded numbers:

Constant	Value
TacmAppWin31	0
TacmAppWin95	1
TacmBdrNormal	0
TacmBdrShadow	2
TacmBdrThreeD	1
TacmBmpBottomToTop	3
TacmBmpCheckerBoard	7
TacmBmpLeftToRight	0
TacmBmpMultipleRect	8
TacmBmpPicture1	1
TacmBmpPicture2	2
TacmBmpRectangle	6
TacmBmpRightToLeft	2
TacmBmpShutterHorizontal	4
TacmBmpShutterVertical	5
TacmBmpTopToBototm	1

Constant	**Value**
TacmCdlCcFullOpen	2
TacmCdlCcPreventFullOpen	4
TacmCdlCcRGBInit	1
TacmCdlCcShowHelp	8
TacmCdlCfAnsiOnly	1024
TacmCdlCfBoth	3
TacmCdlCfEffects	256
TacmCdlCfFixedPitchOnly	16384
TacmCdlCfForceFontExists	65536
TacmCdlCfHelpButton	4
TacmCdlCfLimitSize	8192
TacmCdlCfNoSimulations	4096
TacmCdlCfNoVectorFonts	2048
TacmCdlCfPrinterFonts	2
TacmCdlCfScalableOnly	131072
TacmCdlCfScreenFonts	1
TacmCdlCfTrueTypeOnly	262144
TacmCdlCfWYSIWYG	32768
TacmCdlOfnAllowMultiSelect	512
TacmCdlOfnCreatePrompt	8192
TacmCdlOfnExtensionDifferent	1024
TacmCdlOfnFileMustExist	4096
TacmCdlOfnHelpButton	16
TacmCdlOfnHideReadOnly	4
TacmCdlOfnNoChangeDir	8
TacmCdlOfnNoReadOnlyReturn	32768
TacmCdlOfnNoValidate	256
TacmCdlOfnOverwritePrompt	2
TacmCdlOfnPathMustExist	2048
TacmCdlOfnReadOnly	1
TacmCdlOfnShareAware	16384
TacmCdlPdAllPages	0
TacmCdlPdCollate	16
TacmCdlPdDisablePrintToFile	524288
TacmCdlPdHelpButton	2048
TacmCdlPdHidePrintToFile	1048576
TacmCdlPdNoPageNums	8

Constant	Value
TacmCdlPdNoSelection	4
TacmCdlPdNoWarning	128
TacmCdlPdPageNums	2
TacmCdlPdPrintSetup	64
TacmCdlPdPrintToFile	32
TacmCdlPdReturnDC	256
TacmCdlPdReturnDefault	1024
TacmCdlPdReturnIC	512
TacmCdlPdSelection	1
TacmCdlPdUseDevModeCopies	262144
TacmClk12Hour	0
TacmClk24Hour	1
TacmClkAnalog	1
TacmClkArrow	1
TacmClkDigital	0
TacmClkFilledArrow	2
TacmClkLine	0
TacmClkModern	3
tacmCursorOnHoverDefault	0
tacmCursorOnHoverHyperlinkHand	1
TacmDdsLeft	0
TacmDdsRight	1
TacmDtCustom	3
TacmDtLongDate	0
TacmDtShortDate	1
TacmDtTime	2
TacmEffNormal	0
TacmEffRaised	1
TacmEffSunken	2
TacmEnbBottom	3
TacmEnbLeft	0
TacmEnbRight	2
TacmEnbNormal	0
TacmEnbToggle	1
TacmEnbTop	1
TacmErrCancelError	32755

Constant	Value
TacmErrCantRegister	32754
TacmErrFileNotFound	53
TacmErrInvalidIndexValue	381
TacmErrInvalidPropertyValue	380
TacmErrNotSupported	383
TacmErrNull	94
TacmErrPathFileAccess	75
TacmErrPathNotFound	76
TacmErrPermissionDenied	70
TacmGauArrow	1
TacmGauBottom	3
TacmGauFilledArrow	2
TacmGauFullCircle	1
TacmGauHalfCircle	0
TacmGauLeft	0
TacmGauLine	0
TacmGauModern	3
TacmGauRight	2
TacmGauTickNone	1
TacmGauTickNormal	0
TacmGauTop	1
TacmGauTriangle	4
TacmHlpContents	3
TacmHlpContext	1
TacmHlpContextPopup	8
TacmHlpForceFile	5
TacmHlpHelpCommand	258
TacmHlpHelpOnHelp	4
TacmHlpIndex	3
TacmHlpKey	257
TacmHlpNotepad	0
TacmHlpPartialKey	261
TacmHlpQuit	2
TacmHlpSetContents	5
TacmHlpSetIndex	5
TacmIcmFlowed	2
TacmIcmHorizontal	0

Constant	Value
TacmIcmVertical	1
TacmMrqDown	2
TacmMrqLeft	0
TacmMrqRight	1
TacmMrqUp	3
TacmNteLeft	0
TacmNteTop	1
TacmNteRight	2
TacmNteBottom	3
TacmOriHorizontal	0
TacmOriVertical	1
TacmPrgLeftBottom	0
TacmPrgRightTop	1
TacmPrgSolid	0
TacmPrgVUBar	2
TacmPrgWin95	1
TacmReghKeyClassesRoot	2147483648
TacmReghKeyCurrentConfig	2147483653
TacmReghKeyCurrentUser	2147483649
TacmReghKeyDynData	2147483654
TacmReghKeyLocalMachine	2147483650
TacmReghKeyPerformanceData	2147483652
TacmReghKeyUsers	2147483651
TacmSliBoth	2
TacmSliDirLeftBottom	1
TacmSliDirRightTop	0
TacmSliLeftBottom	0
TacmSliNone	3
TacmSliRightTop	1
TacmSpnNormal	0
TacmSpnThreeD	1
TacmSys386	386
TacmSys486	486
TacmSys586	586
TacmSys8664	8664
TacmSysOther	2

Constant	**Value**
TacmSysCDROM	5
TacmSysFixed	3
TacmSysNoRootDir	1
TacmSysRAMDisk	6
TacmSysRemote	4
TacmSysRemovable	2
TacmSysUnknown	0
TacmSysWin95	0
TacmSysWinNT	1
TacmTabBottom	3
TacmTabClosed	1
TacmTabLeft	0
TacmTabOpen	0
TacmTabRight	2
TacmTabTop	1
TacmTefCenter	0
TacmTefLeft	1
TacmTefTop	2
TacmTefRight	3
TacmTefBottom	4
TacmTxtNormal	0
TacmTxtShadow	2
TacmTxtThreeD	1

Chapter 7: Property, Method, and Event Reference

This chapter lists every property, method, and event that is available to the Total Access Components controls. The items are listed alphabetically and many apply to more than one control. Use the on-line help system and demo files for additional information.

Topics in this Chapter

- **Layout Conventions Used in this Chapter**
- **Property, Method, and Event Reference**

Layout Conventions Used in this Chapter

The information presented in this chapter is shown in three formats depending whether the item is a Property, a Method, or an Event. The following examples show the information that is presented for each type:

Layout Style for Properties

Topic	**Description**		
Applies To	Shows the control(s) for which this property applies.		
Description	Provides a brief description of the property.		
Setting	Provides a description of the valid settings for the property.		
	Setting	Description	VBA
	Setting used in the controls' property sheet.	Brief description of what the setting does.	Value to set the property in code.
Remarks	Lists any notes regarding the property.		

Layout Style for Methods

Topic	**Description**	
Applies To	Shows the control(s) to which this method applies.	
Description	Provides a brief description of what the method does.	
Syntax	Shows the syntax used in VBA to call the method.	
	Part	**Description**
	Parameter to the method.	Description of the parameter.
Remarks	Lists any notes regarding the method.	

Layout Style for Events

Topic	**Description**	
Applies To	Shows the control(s) to which this event applies.	
Description	Provides a brief description of when the event is called.	
Syntax	Shows the syntax used in VBA to declare the event.	
	Argument	**Description**
	Argument that is passed in to the event procedure.	Description of the argument.
Remarks	Lists any notes regarding the event.	

Property, Method, and Event Reference

This section alphabetically lists every property, method, and event available in the Total Access Components custom controls.

AlarmEnable Property

Applies To Clock

Description Enables or disables the OnAlarm event.

Setting The AlarmEnable property has the following settings:

Setting	Description	VBA
Checked	The OnAlarm event fires when it reaches the AlarmTime.	True
Un-Checked	The OnAlarm event does not fire.	False

AlarmTime Property

Applies To Clock

Description The time the OnAlarm event is set to fire.

Setting 5 digit strings with the format “*hh:mm”,* where *hh* specifies the hour and *mm* specifies the minute.

Remarks The default value is “00:00”. AlarmTime is based on the system time, not the clock time.

Alignment Property

Applies To Digital Display

Description The text alignment in the Digital Display.

Setting The Alignment property has the following settings:

Setting	Description	VBA
Left	The text is aligned to the left.	TacmddsLeft
Right	The text is aligned to the right.	TacmddsRight

Angle Property

Applies To Text Effects

Description The angle of the text displayed in the control.

Setting Accepts Integers between 0 and 360.

Remarks To rotate text, you must use a True Type font.

Animate Method

Applies To Bitmap Effects

Description Runs the animation specified by the AnimationEffect property. Specify the number of steps in the animation with the Steps property.

Syntax *ControlName*.Animate

Returns Does not return a value.

AnimationEffect Property

Applies To Bitmap Effects

Description The effect to apply to the bitmaps specified by Picture1 and Picture2.

Setting The AnimationEffect property has the following settings:

Setting	Description	VBA
Left to Right	Animation bitmap starts on the left and pushes initial bitmap to the right.	TacmbmpLeftTo Right
Top to Bottom	Animation bitmap starts at the top and pushes initial bitmap to the bottom.	TacmbmpTopTo Bottom
Right to Left	Animation bitmap starts on the right and pushes initial bitmap to the left.	TacmbmpRightTo Left
Bottom to Top	Animation bitmap starts at the bottom and pushes initial bitmap to the top.	TacmbmpBottom ToTop
ShutterHorizontal	Animation bitmap covers initial bitmap similar to horizontal shutters opening.	TacmBmpShutter Horizontal
ShutterVertical	Animation bitmap covers initial bitmap similar to vertical shutters opening.	TacmBmpShutter Vertical
Rectangle	Animation bitmap covers initial bitmap by drawing a rectangle from the center out.	TacmBmpRectangle
CheckerBoard	Animation bitmap covers initial bitmap by combining horizontal and vertical shutters.	TacmBmpChecker Board
MutipleRect	Animation bitmap covers initial bitmap by drawing multiple rectangles.	TacmBmpMultiple Rect

Appearance Property

Applies To Enhanced Button, Tab

Description The Windows style that the control emulates.

Setting The Appearance property has the following settings:

Setting	Description	VBA
Windows 3.x	Control emulates the Windows 3.x shell.	TacmAppWin31
Windows 95	Control emulates the Windows 95 shell.	TacmAppWin95

ApplicationName Property

Applies To About Box

Description The Application Name shown in the About Box.

Setting String.

ArrowStyle Property

Applies To Spin Button

Description The visual style of the Spin Button arrows.

Setting The ArrowStyle property has the following settings:

Setting	Description	VBA
Normal	Arrows are drawn with a flat effect.	TacmSpnNormal
Three-D	Arrows are drawn with a three-D effect.	TacmSpnThreeD

AutoIncrement Property

Applies To Spin Button

Description Whether the property value is automatically incremented and decremented.

Setting The AutoIncrement property has the following settings:

Setting	Description	VBA
Checked	Value automatically increments and decrements.	True
Un-Checked	Value is not incremented or decremented.	False

BackColor Property

Applies To AVI, Border, Clock, Enhanced Button, Gauge, Digital Display, Icon Menu, Marquee, Notes, Progress Meter, Slider, Spin Button, Splitter, Tab Control, Text Effects

Description The color for the interior of a control.

Setting Long integer.

Remarks On the control's property sheet, select the color in the Color sheet.
In VBA, use a long integer to set this property.

BarColor Property

Applies To Progress Meter

Description The color for the interior the progress bar.

Setting Long integer.

Remarks On the control's property sheet, select the color in the Color sheet.
In VBA, use a long integer to set this property.
BarColor does not apply when BarStyle is set to the Volume Unit (VU) Bar style.

BarStyle Property

Applies To Progress Meter

Description The display style of the progress meter.

Setting The BarStyle property has the following settings:

Setting	Description	VBA
Solid	Progress bar is solid, like in Windows 3.1	TacmPrgSolid
Windows 95	Progress bar is segmented, like in Windows 95.	TacmPrgWin95
VU Bar	Progress bar is segmented like in Windows 95, but supports multiple colors in the bar.	TacmPrgVUBar

Remarks The caption property only applies if this property is set to Solid.

BevelWidth Property

Applies To Border

Description The bevel's visual effect.

Setting Integer.

Remarks Applies only if the EffectStyle is set to Beveled.

BodyStyle Property

Applies To Tab Control

Description The display style of the body of the tab.

Setting The BodyStyle property has the following settings:

Setting	Description	VBA
Open	The body is open.	TacmTabOpen
Closed	The body is closed.	TacmTabClosed

Remarks This property is used to simulate containership. Refer to page 85 for details.

BorderColor Property

Applies To Border

Description The color for the surrounding border of a control.

Setting Long integer.

Remarks On the control’s property sheet, select the color in the Color sheet.
In VBA, use a long integer to set this property.

BrowseForComputer Property

Applies To Browse for Folder

Description True to limit selection to a computer, or false to allow folder/file selection as well.

Setting Boolean.

Remarks This property only works when the EditBox property is False.

BrowseForPrinter Property

Applies To Browse for Folder

Description True to limit selection to a printer, or false to allow folder/file selection.

Setting Boolean.

BrowseIncludeFiles Property

Applies To Browse for Folder

Description True to allow file selection, or False to limit selection to the folder level.

Setting Boolean

ButtonStyle Property

Applies To Enhanced Button

Description The style of the button: Normal or Toggle.

Setting The ButtonStyle property has the following settings:

Setting	Description	VBA
Normal	Button behaves like a standard Windows button.	TacmEnbNormal
Toggle	The button behaves like a toggle button.	TacmEnbToggle

CancelError Property

Applies To Common Dialog

Description Whether an error is generated when the user chooses the [Cancel] button.

Setting The CancelError property has the following settings:

Setting	Description	VBA
Checked	An error is generated when [Cancel] is clicked.	True
Un-Checked	No error is generated when [Cancel] is clicked.	False

Remarks If True, error number 32755 occurs when the user clicks [Cancel] on the common dialog. Trap for this error to respond appropriately.

CapsLock Property

Applies To System Information

Description The state of the [Caps Lock] key.

Setting This property has the following settings:

Description	VBA
CapsLock is on.	True
CapsLock is off.	False

Remarks This property is read-only. Assigning a value to this property triggers an error.

Caption Property

Applies To Enhanced Button, Digital Display, Progress Meter

Description The text displayed in the control.

Setting String.

Remarks **Enhanced Button**: The text to display when the button is in the up state.

Digital Display: The text that displayed in digital format. If a character cannot be displayed, it is shown as an **8**.

Progress Meter: The text to display when BarStyle is set to Solid.
"%"is replaced with "*x*%," (where *x* is the current value).

CaptionDown Property

Applies To Enhanced Button

Description The text displayed in the control when the button is toggled down.

Setting String.

Clear Method

Applies To Clipboard

Description Clears the contents of the clipboard.

Syntax *ControlName*.Clear

Returns Does not return a value.

ClipPicture Property

Applies To	Clock
Description	True to clip the picture to the clock’s circle, or False to fit the control.
Setting	The ClipPicture property has the following settings:

Setting	Description	VBA
Checked	Picture is clipped to fit the clock’s circle.	True
Un-Checked	Picture is sized to fit the clock control.	False

Remarks This property only applies when the clock is analog.

ClockStyle Property

Applies To	Clock
Description	The appearance of the clock face: Digital or Analog.
Setting	The ClockStyle property has the following settings:

Setting	Description	VBA
Digital	Displays a digital interface.	TacmClkDigital
Analog	Displays an analog interface.	TacmClkAnalog

Close Method

Applies To	AVI, CD Player, Resize
Description	Closes the resource.
Syntax	*ControlName*.Close
Returns	Does not return a value.

Color Property

Applies To	Common Dialog
Description	The color selected .
Setting	Long integer.
Remarks	This property reads or sets the color in the font or color dialog.

ComputerName Property

Applies To	System Information
Description	The network computer name.
Setting	String.
Remarks	This property is read-only. Assigning a value to this property causes an error. This property is not available unless a network is installed.

Copies Property

Applies To	Common Dialog
Description	The number of copies to be printed.
Setting	Integer.
Remarks	The Copies property is run-time only.

For the Print dialog box, this property returns the number of copies entered by the user in the Copies box. If the flag is set to 262144 for the Common Dialog control, this property always returns 1.

For the Printer object, multiple copies may or may not be collated, depending on the printer driver. Multiple copies of the entire document or multiple copies of each page may be printed. For printers that don't support collating, set Copies = 1, and then use a loop in code to print multiple copies of the entire document.

Copyright Property

Applies To	About Box
Description	The Copyright notice shown in the About Box.
Setting	String.

CPUType Property

Applies To	System Information
Description	The type of CPU on the computer.
Setting	The CPUType property has the following settings:

Setting	Description	VBA
386	386 Processor.	TacmSys386
486	486 Processor.	TacmSys486
586	Pentium or higher Processor.	TacmSys586
8664	64-Bit Processor.	TacmSys8664
N/A	Processor type is unrecognized.	TacmSysOther

Remarks This property is read-only.

CurrentPicture Property

Applies To	Bitmap Effects
Description	The picture currently shown in the control: Picture1 or Picture2.
Setting	The CurrentPicture property has the following settings:

Setting	Description	VBA
1	Control shows Picture1.	TacmBmpPicture1
2	Control shows Picture2.	TacmBmpPicture2

Remarks Setting this property causes the picture displayed to switch instantly.

CurrentStep Property

Applies To	Bitmap Effects
Description	The step in the animation that is currently displayed.

Setting Integer between 0 and the Steps property.

CursorOnHover Property

Applies To Enhanced Button

Description The appearance of the cursor when hovering on the button.

Setting The CursorOnHover has the following settings:

Value	Description	Example
0	Show default cursor when hovering .	tacmCursorOnHoverDefault
1	Show HyperlinkHand cursor when hovering.	tacmCursorOnHoverHyperlinkHand

CurveHeight Property

Applies To Border

Description The amount of y curvature for the border control.

Setting Integer.

Remarks This property only applies if the EffectStyle is Normal.

Setting CurveHeight and CurveWidth to zero draws a rectangle.

CurveWidth Property

Applies To Border

Description The amount of x curvature for the border control.

Setting Integer.

Remarks This property only applies if the EffectStyle is Normal.

Setting CurveHeight and CurveWidth to zero draws a rectangle.

CustomFormat Property

Applies To DateTime Picker

Description The format to use when the Format property is Custom.

Setting String with the following values:

Value	Description	Example
d	One- or two-digit day.	May **1** or May **10**
dd	Two-digit day.	May **01** or May **10**
ddd	Three-character weekday abbreviation.	Wed
dddd	Full weekday name.	Wednesday
h	One- or two-digit hour in 12-hour format.	**9**:00 or **12**:00
hh	Two-digit hour in 12-hour format.	**09**:00 or **12**:00
H	One- or two-digit hour in 24-hour format.	**9**:00 or **18**:00
HH	Two-digit hour in 24-hour format.	**09**:00 or **18**:00
m	One- or two-digit minute.	09: **5** or 09:**15**
mm	Two-digit minute.	09:**05** or 09:**15**
M	One- or two-digit month number.	**1**/14 or **10**/14
MM	Two-digit month number.	**01**/14 or **10**/14
MMM	Three-character month abbreviation.	Jan
MMMM	Full month name.	January
t	One-letter AM/PM abbreviation.	1:15 **A** or 1:15 **P**
tt	Two-letter AM/PM abbreviation.	1:15 **AM** or 1:15 **PM**
yy	Last two digits of the year.	Jan. 14, **04**
yyyy	Four-digit year.	Jan. 14, **2004**

Remarks This property only applies if the Format property is Custom.

DefaultExt Property

Applies To Common Dialog

Description The default extension displayed in File Open and File Save Common Dialogs.

Setting String containing any valid file extension.

Remarks This property specifies a default file extension (such as .TXT or .DOC). When a file with no extension is saved, this extension is appended to the filename.

Delay Property

Applies To Bitmap Effects

Description The delay, in milliseconds, between steps in an animation.

Setting Integer.

Remarks There are 1000 milliseconds in a second.

DeleteKey Method

Applies To Registry Control

Description Delete a registry key.

Syntax *ControlName*.DeleteKey *RootKey*, *Key*

Part	Description
RootKey	Long integer that specifies the root key of the key to delete.
Key	String that specifies the key to delete.

Returns This method does not return a value.

Remarks Be very careful when using this method—deleting an incorrect key may render Windows unusable.

DeleteKeyValue Method

Applies To Registry Control

Description Delete a value.

Syntax *ControlName*.DeleteKeyValue *RootKey*, *Key*, *Value*

Part	Description
RootKey	Long integer that specifies the root key of the key to delete.
Key	String that specifies the key to delete.
Value	String that specifies the key value to delete.

Returns This method does not return a value.

Remarks Be very careful when using this method—deleting an incorrect key may render Windows unusable.

DialogTitle Property

Applies To About Box, Common Dialog

Description The title of the About Box or Common Dialog control.

Setting String.

Remarks The default for an Open dialog is “Open”, the default for a Save As dialog is “Save As”, and the default for the About Box is “About Box”.

This property doesn’t apply to the Color and Font common dialogs.

Direction Property

Applies To Marquee, Slider

Description The direction that the marquee text moves, or the direction that the slider increases.

Setting The Direction has the following settings:

	Setting	Description	VBA
Marquee	Left	Text scrolls towards the left.	TacmMrqLeft
	Right	Text scrolls towards the right.	TacmMrqRight
	Down	Text scrolls towards the bottom.	TacmMrqDown
	Up	Text scrolls towards the top.	TacmMrqUp
Slider	Right/Top	Slider increases towards the right or top.	TacmSliDirRightTop
	Left/Bottom	Slider increases towards the left or bottom.	TacmSliDirLeftBottom

Remarks The windows standard is for a slider to increase to the right if it is horizontal, and to the bottom if it is vertical.

DontGoBelowDomain Property

Applies To Browse for Folder

Description Allow or disallow selecting a network computer.

Setting Boolean.

DoPaint Method

Applies To Text Effects

Description Repaint the control.

Syntax *ControlName*.DoPaint

Returns Does not return a value.

Remarks This method is useful if you are changing a property faster than the control can repaint itself, such as setting the Angle property in a For loop.

EditBox Property

Applies To Browse for Folder

Description True to display a field for typing a path, or False to force the user to use the treeview.

Setting Boolean.

EffectOffset Property

Applies To Clock, Enhanced Button, Marquee, Text Effects

Description Separation distance, in pixels, between the highlight text, display text, and shadow text.

Setting Integer.

EffectStyle Property

Applies To Border, Clock, Gauge, Progress Meter

Description The appearance of the border around the control.

Setting The EffectStyle property has the following settings:

Border

Setting	Description	VBA
Normal	Standard border.	TacmBdrNormal
Three-D	Three dimensional border.	TacmBdrThreeD
Shadow	Shadow border.	TacmBdrShadow

Clock, Gauge, Progress Meter

Setting	Description	VBA
Normal	Standard border.	TacmEffNormal
Raised	Raised border.	TacmEffRaised
Sunken	Sunken border.	TacmEffSunken

Eject Method

Applies To CD Player

Description Open the CD-ROM drive door.

Syntax *ControlName*.Eject

Returns Does not return a value.

Enabled Property

Applies To Cursor, Enhanced Button, Icon Menu, Slider, Spin Button, Tab Control

Description Whether an object can respond to user-generated events.

Setting The Enabled property has the following settings:

Setting	Description	VBA
Checked	Control responds to user interaction.	True
Un-Checked	Control does not respond to user interaction.	False

Remarks A control still receives input focus when it is not enabled, but it doesn’t respond to user interaction.

EnableTheme Property

Applies To Gauge, Notes, Popup Menu

Description Whether the control respects the Windows Theme

Setting Boolean (True to respect the Windows Theme)

Remarks When EnableTheme is True, certain appearance properties (such as BackColor) are not respected. Instead, the appearance is inherited form the Windows Theme.

FileName Property

Applies To About Box, AVI, Common Dialog, INI

Description The full path and name of a selected file.

Setting String.

Remarks When you create the control at run-time, FileName is a zero length string (“”), meaning no file is selected.

Common Dialog: Optionally set FileName to an existing file before opening the dialog to set the initial FileName. Read this property to return the currently selected filename. If no file is selected, FileName returns a zero length string.

INI: If FileName is set to a non-existent file, the file is created when you call SetValue.

FileTitle Property

Applies To Common Dialog

Description The name (without the path) of the file to open or save.

Setting String.

Remarks FileTitle is set when the user selects a file and clicks [OK] in the Open or Save dialog, and can be used to open or save the selected file.

If the flag is set to 256, the FileTitle property doesn’t return a value.

FillStyle Property

Applies To Progress Meter.

Description The location where the progress bar begins to fill.

Setting The FillStyle property has the following settings:

Setting	Description	VBA
Left/Bottom	Progress bar fills from the left or bottom.	TacmPrgLeftBottom
Right/Top	Progress bar fills from the right or top.	TacmPrgRightTop

Filter Property

Applies To Common Dialog

Description The filters that are displayed in the Type list box of the dialog box.

Setting Filter = *Description1* |*Filter1* |*Description2* |*Filter2*…
The Filter property syntax has these parts:

Part	Description
Description	String that describes the type of file.
Filter	String that specifies the filename extension.

Remarks Filter specifies the type of files that are displayed in the dialog box’s file list. For example, selecting the filter *.TXT displays text files.

Use this property to provide the user with a list of files that can be selected or saved in the dialog. Use the pipe (|) symbol to separate the description and filter values.

When you specify more than one filter for a dialog box, use the FilterIndex property to set the default filter.

FilterIndex Property

Applies To	Common Dialog
Description	The index for the default filter in the Open or Save File dialog.
Setting	Integer.
Remarks	Use the Filter property to set filters. The index for the first filter is 1.

Flags Property

Applies To	Common Dialog
Description	Various options for the dialog boxes. Since this is a very complicated property, please refer to this documentation along with the sample database for information and examples.
Setting	The Flags property has the following settings:

Color Dialog

Description	VBA
Displays the entire dialog box, including the portion that allows custom colors, when the dialog box shows.	TacmCdlCcFullOpen
Disables the [Define Custom Colors] button, and prevent users from defining custom colors.	TacmCdlCcPreventFullOpen
Sets the Initial Color value for the dialog box.	TacmCdlCcRGBInit
Displays the [Help] button.	TacmCdlCcHelpButton

File Open/ Save Dialog

Description	VBA
Prompts the user to create a file if it does not exist.	TacmCdlOfnCreatePrompt
Indicates that the extension of the returned filename is different from the DefaultExt.	TacmCdlOfnExtensionDifferent
Limits users to entering existing file names in FileName.	TacmCdlOfnFileMustExist
Hides the Read Only check box.	TacmCdlOfnHideReadOnly
Forces the dialog to set the current directory to what it was when the dialog box was opened.	TacmCdlOfnNoChangeDir
Specifies that the returned file won’t be Read Only or in a write-protected folder.	TacmCdlOfnReadOnly
Does not validate the returned FileName, so it may include invalid characters.	TacmCdlOfnNoValidate
In the Save dialog, prompts the user to confirm overwriting the file if the selected file exists.	TacmCdlOfnOverWritePrompt
Limits the user to entering valid paths. If the user enters an invalid path, displays a warning.	TacmCdlOfnPathMustExist

Checks the Read Only check box by default.	TacmCdlOfnReadOnly
Ignores sharing violation errors.	TacmCdlOfnShareAware
Displays the [Help] button.	TacmCdlOfnHelpButton

Font Dialog

Description	VBA
Lists the available printer and screen fonts.	TacmCdlCfBoth
Enables strike through, underline, and color effects.	TacmCdlCfEffects
Displays only fixed-pitch fonts.	TacmCdlCfFixedPitchOnly
Displays an error message if the user selects a font or style that doesn't exist.	TacmCdlCfFontExists
Displays only font sizes within the range specified by the Min and Max properties.	TacmCdlCfLimitSize
Disallows Graphic Device Interface (GDI) font simulation.	TacmCdlCfNoSimulations
Disallows vector-font selections.	TacmCdlCfNoVectorFonts
Lists only the fonts supported by the printer.	TacmCdlCfPrinterFonts
Allows selecting only scalable fonts.	TacmCdlCfScalableOnly
Lists only screen fonts supported by the system.	TacmCdlCfScreenFonts
Allows selecting only TrueType fonts.	TacmCdlCfTTOnly
Allows selecting only fonts that are available on both the printer and screen.	TacmCdlCfWYSIWYG
Displays the [Help] button.	TacmCdlCfHelpButton

Print/Print Setup Dialog

Description	VBA
Returns or sets the Collate check box.	TacmCdlPdCollate
Disables the Print To File check box.	TacmCdlPdDisablePrintToFile
Hides the Print To File check box.	TacmCdlPdHidePrintToFile
Disables the Pages option and associated controls.	TacmCdlPdNoPageNums
Disables the Selection option.	TacmCdlPdNoSelection
Prevents a warning message from being displayed when there is no default printer.	TacmCdlPdNoWarning
Returns or sets the Pages option button.	TacmCdlPdPageNums
Displays the Print Setup dialog box rather than the Print dialog box.	TacmCdlPdPRintSetup
Returns or sets the Print To File check box.	TacmCdlPdPrintToFile
Returns a device context for the printer selection. The device context is the hDC property.	TacmCdlPdReturnDC
Returns an information context for the printer selection. The information context is returned in the hDC property.	TacmCdlPdReturnIC

The default printer name.	TacmCdlPdReturnDefault
The Selection option button.	TacmCdlPdSelection
If a printer driver doesn’t support multiple copies: disables the copies edit control. If a driver does support multiple copies: Indicates that the dialog box stores the number of copies set in the Copies property.	TacmCdlPdUseDevModeCopies
Displays the [Help] button.	TacmCdlPdHelpButton

Remarks Use the on-line help and the sample for more information on using this property.

Font Property

Applies To About Box, Clock, Enhanced Button, Icon Menu, Marquee, Notes, Progress Meter, Tab Control, Text Effects

Description Information needed to format text for display or for printed output.

Setting The Font property has the following elements:

Item	Description	VBA
Bold	Text is bold (True/False).	Font.Bold
Italic	Text is italicized (True/False).	Font.Italic
Name	Font Name.	Font.Name
Size	Font size. See **Font Size Adjustments** on page 37 for details.	Font.Size
StrikeThrough	Text has a line drawn through it (True/False).	Font.StrikeThru
Underline	Text is underlined (True/False).	Font.Underline

Remarks This property is a font object containing properties of its own. Refer to the properties of the font by using the VBA settings listed (e.g. Font.Bold).

Icon Menu: This property is obsolete for the Icon Menu control (replaced by ItemFont).

FontBold Property

Applies To Common Dialog

Description The font’s bold property in the Font Common Dialog.

Setting Boolean.

FontItalic Property

Applies To Common Dialog

Description The font’s italic style in the Font Common Dialog.

Setting Boolean.

FontName Property

Applies To Common Dialog

Description The name of the font in the Font Common Dialog.

Setting String containing the name of any Windows font.

Remarks If you select a font that your system can’t display or that isn’t installed, Windows substitutes a similar font.

FontSize Property

Applies To	Common Dialog
Description	The point size of the font in the Font Common Dialog.
Setting	Integer.
Remarks	A point is 1/72 of an inch.

FontStrikeOut Property

Applies To	Common Dialog
Description	The font's strikeout style in the Font Common Dialog.
Setting	Boolean.

FontUnderline Property

Applies To	Common Dialog
Description	The font's underline style in the Font Common Dialog.
Setting	Boolean.

ForeColor Property

Applies To Clock, Enhanced Button, Digital Display, Gauge, Icon Menu, Marquee, Notes, Spin Button, Text Effects

Description The color for the Foreground of a control.

Setting Long integer.

Remarks On the control's property sheet, select the color in the Color sheet. In VBA, use a long integer to set this property.

Clock: Determines the font color if ClockStyle is set to Digital, or the color of the clock hands and tick marks if it's Analog.

Enhanced Button, Digital Display, and Text Effects: Determines the color of the text displayed in the control.

Gauge: Determines the color of the needle and tick marks.

Icon Menu: This property is obsolete for the Icon Menu control (replaced by ItemForeColor).

Format Property

Applies To DateTime Picker

Description Format of the date/time displayed.

Setting The Format property has the following settings:

Setting	Description	VBA
Custom	Use the format specified in CustomFormat.	TacmDtCustom
Long Date	Use the system's long date format.	TacmDtLongDate
Short Date	Use the system's short date format.	TacmDtShortDate
Time	Use the system's time format.	TacmDtTime

FreeMem Property

Applies To System Information

Description Total free physical memory, in bytes.

Setting Long integer.

Remarks This property is retained for backward compatibility only. Use FreeMemEx to support larger values.

This property is read-only. Assigning a value to this property triggers an error.

FreeMemEx Property

Applies To System Information

Description Total free physical memory, in bytes. Supports larger values than FreeMem.

Setting Double.

Remarks This property is read-only. Assigning a value to this property triggers an error.

FromPage Property

Applies To Common Dialog

Description The value for the From text box in the Print dialog.

Setting Integer.

Remarks Return or set the values for the From and To text boxes. These properties are valid only when the flag is set to 8.

GaugeStyle Property

Applies To Gauge

Description The visual style of the Gauge: half or full circle.

Setting The GaugeStyle property has the following settings:

Setting	Description	VBA
Half-Circle	Shaped like the top half of a circle.	TacmGauHalfCircle
Full-Circle	Shaped like a full circle.	TacmGauFullCircle

GetDiskType Method

Applies To System Information

Description Returns the type of the specified drive.

Syntax *Integer* = *ControlName*.GetDiskType(*Drivenumber*)

Part	Description
Drivenumber	Integer specifying the 0-based drive number.

Returns Integer.

Remarks The *drivenumber* parameter is interpreted as follows:

0 - A drive, 1 - B drive, 2 - C drive, … 25 – Z drive

GetDiskVolume Method

Applies To System Information

Description Returns the volume name of the specified drive.

Syntax *String*= *ControlName*.GetDiskVolume(*Drivenumber*)

Part	Description
Drivenumber	Integer specifying the 0-based drive number.

Returns String.

Remarks The *drivenumber* parameter is interpreted as follows:

0 - A drive, 1 - B drive, 2 - C drive, ... 25 – Z drive

GetFileExt Method

Applies To System Information

Description Returns the file extension.

Syntax *String* = *ControlName*.GetFileExt(*Filename*)

Part	Description
Filename	String that specifies the file with the extension to retrieve.

Returns String.

GetFileName Method

Applies To System Information

Description Returns the file title portion of the path, without the file path or extension.

Syntax *String* = *ControlName* GetFileName(*Filename*)

Part	Description
Filename	String that specifies the file with the file name to retrieve.

Returns String.

GetFilePath Method

Applies To System Information

Description Returns the path of the file, without the filename or extension.

Syntax *String* = *ControlName*.GetFilePath(*Filename*)

Part	Description
Filename	String that specifies the file with the path to retrieve.

Returns String.

GetFileVersion Method

Applies To System Information

Description Returns the subkey string specified by the index and key parameters.

Syntax *String* = *ControlName*.GetFileVersion(*Path*)

Part	Description
Path	String that specifies the path to the file with the version you want to retrieve.

Returns This method returns a string.

Remarks If the path is invalid or the file doesn’t have a version, returns a zero length string.

GetFreeDiskSpace Method

Applies To System Information

Description Returns the amount of free disk space on the specified drive.

Syntax *LongInteger* = *ControlName*.GetFreeDiskSpace(*Drivenumber*)

Part	Description
Drivenumber	Integer specifying the 0-based drive number.

Returns Long integer.

Remarks This method is retained for backward compatibility only. Use GetFreeDiskSpaceEx to support larger values.

The *drivenumber* parameter is interpreted as follows:

0 - A drive, 1 - B drive, 2 - C drive, ... 25 – Z drive

GetFreeDiskSpaceEx Method

Applies To System Information

Description Returns the amount of free disk space on the specified drive.

Syntax *LongInteger* = *ControlName*.GetFreeDiskSpaceEx(*Drivenumber*)

Part	Description
Drivenumber	Integer specifying the 0-based drive number

Returns Double.

Remarks This method replaces GetFreeDiskSpace, and supports larger values.

The *drivenumber* parameter is interpreted as follows:

0 - A drive, 1 - B drive, 2 - C drive, ... 25 – Z drive

GetKeyValue Method

Applies To Registry

Description Reads the value of the specified key.

Syntax *String* = *ControlName*.GetKeyValue(*RootKey*, *Key*, *Value*)

Part	Description
RootKey	Long integer that specifies the root key of the key to retrieve.
Key	String that specifies the key to retrieve.
Value	String that specifies the key value to retrieve.

Returns Returns a variant.
To retrieve the default data for a key, pass a blank string (“”) to the value.

GetPicture Method

Applies To Clipboard

Description Reads an image from the clipboard.

Syntax *Variant* = *ControlName*.GetPicture

Returns Variant.

Remarks This method returns an empty variant if the clipboard is empty.

GetSubkeyCount Method

Applies To Registry

Description Returns the number of child keys of the specified key.

Syntax *Integer* = *ControlName*.GetSubkeyCount(*RootKey*, *Key*)

Part	Description
RootKey	Long integer that specifies the root key of the key to retrieve.
Key	String that specifies the key to retrieve.

Returns Integer.

GetSubkeyString Method

Applies To Registry

Description Returns the subkey string specified by the index and key parameters.

Syntax *String* = *ControlName*.GetSubkeyString (*RootKey*, *Key*, *Index*)

Part	Description
RootKey	String that specifies the root key to retrieve.
Key	String that specifies the top-level key to retrieve.
Index	Zero based integer that specifies the sub key to retrieve.

Returns String.

GetText Method

Applies To Clipboard

Description Reads text from the clipboard.

Syntax *String* = *ControlName*.GetText

Returns String.

Remarks This method returns a zero length string if the clipboard is empty.

GetValue Method

Applies To INI Control

Description Reads a value from an INI file.

Syntax *String* = *ControlName.*GetValue(*Section*, *Entry*, *Default*)

Part	Description
Section	String that specifies the initialization file's section heading.
Entry	String that specifies the entry to retrieve.
Default	Default value to return if the entry is not found.

Returns String.

Remarks This method uses the file specified in the Filename property.

GetValueCount Method

Applies To Registry

Description Returns the number of values for the specified key.

Syntax *Integer* = *ControlName*.GetValueCount(*RootKey*, *Key*)

Part	Description
RootKey	Long integer that specifies the root key of the key to retrieve.
Key	String that specifies the key to retrieve.

Returns Integer.

GetValueString Method

Applies To Registry

Description Returns the subkey string specified by the index and key parameters.

Syntax *String* = *ControlName*.GetValueString (*RootKey*, *Key*, *Index*)

Part	Description
RootKey	String that specifies the root key of the key to retrieve.
Key	String that specifies the top-level key to retrieve.
Index	Zero based integer that specifies the sub key to retrieve.

Returns String.

HandStyle Property

Applies To Clock

Description The display style of the clock hands.

Setting The HandStyle property has the following settings:

Setting	Description	VBA
Line	Clock hands display as lines.	TacmClkLine
Arrow	Clock hands display as arrows.	TacmClkArrow
Filled Arrow	Clock hands display as filled arrows.	TacmClkFilledArrow
Modern	Clock hands show as a modern arrow.	TacmClkModern

Remarks This property only applies when the clock is analog.

hDC Property

Applies To Common Dialog

Description A handle provided by the Microsoft Windows operating environment to the device context of an object.

Setting Allows access to a long integer representing a device context.

Remarks This property returns a device context for the printer selected in the Print dialog box.

HelpCaption Property

Applies To About Box

Description The text shown in the [Help] button.

Setting String.

Remarks The HelpCaption supports mnemonics, e.g. “&Help” is displayed as “Help”.

HelpCommand Property

Applies To About Box, Common Dialog

Description The type of on-line Help requested.

Setting The settings for this property are:

Description	VBA

Displays Notepad and opens the file in the HelpFile property. Note that only text files display in Notepad.	TacmHlpNotepad
Displays Help for a particular topic.	TacmHlpContext
Notifies Help application that the Help file is no longer in use.	TacmhlpQuit
Displays the index of the specified Help file.	TacmHlpIndex
Displays the contents topic in the current Help file.	TacmHlpContents
Displays Help for using the Help application itself.	TacmHlpHelpOnHelp
Sets the current index for multi-index Help.	TacmHlpSetIndex
Designates a specific topic as the contents topic.	TacmHlpSetContents
Displays a topic identified by a context number.	TacmHlpContextPopup
Creates a Help file that displays text in only one font.	TacmHlpForceFile
Displays Help for a particular keyword.	TacmHlpKey
Displays Help for a particular command.	TacmHlpCommandhelp
Calls the search engine in Windows Help.	TacmHlpPartialKey

HelpContext Property

Applies To About Box, Common Dialog

Description The context ID of the requested Help topic.

Setting Integer.

Remarks Use this property with the HelpCommand property (set HelpCommand = cdlHelpContext) to specify the Help topic.

HelpEnable Property

Applies To About Box

Description Whether to enable the [Help] button on the dialog box.

Setting Boolean.

HelpFile Property

Applies To About Box, Common Dialog.

Description Path and filename of a Microsoft Windows help (HLP) or Compressed HTML Help (CHM) file to display.

Setting String.

HelpKey Property

Applies To About Box, Common Dialog

Description Keyword that identifies the Help topic.

Setting String.

Remarks Use this property with the HelpCommand property.

HighlightColor Property

Applies To Border, Clock, Enhanced Button, Marquee, Spin Button, Text Effects

Description The color for the highlighted portions of a control.

Setting Long integer.

Remarks On the control’s property sheet, select the color in the Color sheet.
In VBA, use a long integer to set this property.
This property is only applicable when the special effect is set to Three-D.

Init Method

Applies To Resize

Description Initializes the Resize control.

Syntax *ControlName*.Init

Returns Does not return a value.

Remarks This method initializes the resize ratios for the form.

InitDir Property

Applies To Common Dialog

Description The initial file folder for the common dialog.

Setting String.

Remarks If this property isn’t specified, the current folder is used.

Interval Property

Applies To Marquee, Timer

Description **Marquee:** Interval, in milliseconds, between steps.
Timer: Interval, in milliseconds, between OnTimer events.

Setting Integer.

ItemBackColor(i) Property

Applies To Icon Menu, Popup Menu

Description The background color for the specified menu item.

Setting Long integer.

Remarks On the control’s property sheet, select the color in the Color sheet.
In VBA, use a long integer to set this property.

ItemCount Property

Applies To Icon Menu, Popup Menu

Description The number of menu items.

Setting Integer greater than 0.

ItemChecked(i) Property

Applies To Popup Menu

Description Check or un-check a popup menu item.

Setting Boolean.

Remarks This item is a zero based property array. Use an index number from 0 to ItemCount - 1.

ItemEnabled(i) Property

Applies To Popup Menu

Description Enable or disable a menu item.

Setting Boolean.

Remarks This item is a zero based property array. Use an index number from 0 to ItemCount - 1.

ItemFont(i) Property

Applies To Icon Menu, Popup Menu

Description The font for the specified menu item. Font contains information needed to format text for display or printed output.

Setting The Font property has the following elements:

Item	Description	VBA
Bold	Text is bold (True/False).	Font.Bold
Italic	Text is italicized (True/False).	Font.Italic
Name	Font Name.	Font.Name
Size	Font size. See **Font Size Adjustments** on page 37 for details.	Font.Size
StrikeThrough	Text has a line drawn through it (True/False).	Font.StrikeThru
Underline	Text is underlined (True/False).	Font.Underline

Remarks This property is a font object containing properties of its own. Refer to the properties of the font by using the VBA settings listed (e.g. Font.Bold).

ItemForeColor(i) Property

Applies To Icon Menu, Popup Menu

Description The text color for the specified menu item.

Setting Long integer.

Remarks On the control's property sheet, select the color in the Color sheet.
In VBA, use a long integer to set this property.

ItemIcon(i) Property

Applies To Icon Menu, Popup Menu

Description The graphic to display for the item.

Setting The ItemIcon property is only available at design time.

Remarks To load a picture at run-time, use the LoadIcon method.

ItemSeparator(i) Property

Applies To Popup Menu

Description Displays the menu item as a Separator line when true, or as a standard menu item when false.

Setting Boolean.

Remarks This item is a zero based property array. Use an index number from 0 to ItemCount - 1.

ItemText(i) Property

Applies To Icon Menu, Popup Menu

Description The text displayed in a menu item.

Setting String.

Remarks This item is a zero based property array. Use an index number from 0 to ItemCount - 1.

LargeChange Property

Applies To Slider, Spin Button

Description The number of units the Value property changes by when a large change happens.

Setting Integer.

Remarks Default value is 5.

Slider: Large change event occurs when the mouse is clicked to the left or right of the Value marker, or when [Page Up] or [Page Down] is pressed.

Spin: Large change event occurs when the mouse is clicked on one of the arrows with the [CTRL] key held down.

LargeIcons Property

Applies To Icon Menu

Description Enable or disable large icons in the menu.

Setting Boolean. (True to display large icons, and False to display standard icons.)

Length Property

Applies To AVI

Description The length, in milliseconds, of the video clip.

Setting Long integer.

Remarks This property is read-only. Assigning a value to this property causes an error.

LicensedTo Property

Applies To About Box

Description The “Licensed To” text displayed in the About Box.

Setting String.

LicenseString Property

Applies To About Box

Description The licensing rules (or copyright information) text show in the about box.

Setting String.

LineWidth Property

Applies To Gauge

Description The thickness of the lines used in the control.

Setting Integer.

Example *ControlName*.LineWidth = 2

LoadIcon Method

Applies To Icon Menu, Popup Menu

Description Loads a picture from disk into the ItemIcon property.

Syntax *Boolean* = *ControlName*.LoadIcon(*Index*, *Filename*)

Part	Description
Index	The index of the icon item.
Filename	The path and filename of the picture to load.

Returns Boolean.

Remarks This method loads pictures into the IconItem property at runtime. If this method fails, False is returned; otherwise True is returned.

Passing a zero length (“”) string clears the Picture property.

LoadPicture Method

Applies To About Box, Clock, Cursor, Enhanced Button, Gauge

Description Loads a picture from disk into the Picture property.

Syntax *Boolean* = *ControlName*.LoadPicture(*Filename*)

Part	Description
Filename	The path and filename of the picture to load.

Returns Boolean.

Remarks This method loads pictures into the Picture property at run time. If this method fails, False is returned; otherwise True is returned.

Passing a zero length string (“”) clears the Picture property.

LoadPicture1 Method

Applies To Bitmap Effects

Description Loads a picture from disk into the Picture1 property.

Syntax *Boolean* = *ControlName*.LoadPicture1(*Filename*)

Part	Description
Filename	The path and filename of the picture to load.

Returns Boolean.

Remarks This method loads pictures into the Picture1 property at run time. If this method fails, False is returned; otherwise True is returned.

Passing a zero length string (“”) clears the Picture1 property.

LoadPicture2 Method

Applies To Bitmap Effects

Description Loads a picture from disk into the Picture2 property.

Syntax *Boolean* = *ControlName*.LoadPicture2(*Filename*)

Part	Description
Filename	The path and filename of the picture to load.

Returns Boolean.

Remarks This method loads pictures into the Picture2 property at run time. If this method fails, False is returned; otherwise True is returned.
Passing a zero length string (“”) clears the Picture2 property.

LoadPictureDown Method

Applies To Enhanced Button

Description Loads a picture from disk into the PictureDown property.

Syntax *Boolean* = *ControlName*.LoadPictureDown(*Filename*)

Part	Description
Filename	The path and filename of the picture to load.

Returns Boolean.

Remarks This method loads pictures into the PictureDown property at run time. If this method fails, False is returned; otherwise True is returned.
Passing a zero length string (“”) clears the PictureDown property.

LongDate Property

Applies To Clock

Description Enable or disable display of the long date format.

Setting Boolean. (True to display long dates, and False to display short dates.)

MathCo Property

Applies To System Information

Description Whether the system has a math co-processor.

Setting Boolean. (True if a math co-processor is present, and False if it is not.)

Remarks This property is run-time and read-only. Assigning a value causes an error.

Max Property

Applies To Common Dialog, Gauge, Progress Meter, Slider, Spin Button, Splitter

Description The maximum value of the control.

Setting Integer.

Remarks The Max and Min properties define the range of a control. Assigning a value lower than the Min value results in an error.
Font Common Dialog: The flag must be set to 8192 before using this property.
Print Common Dialog: The largest number of pages allowed in the To field.
Gauge, Progress Meter, Slider, and Spin Button: The maximum number that the Value property can take.
Splitter: The maximum value (in twips) where the splitter can move.

MaxFileSize Property

Applies To Common Dialog

Description The maximum size of the file opened using the Common Dialog Control.

Setting Integer between 1 and 256.

Remarks Default value is 256.

MaxHeight Property

Applies To Resize

Description The maximum height of the form.

Setting Long integer.

Remarks MaxHeight is expressed in twips. 1 inch is 1440 twips.

MaxText Property

Applies To Notes

Description The maximum size of the text the user can enter into the Notes popup window.

Setting Integer between 1 and 32K.

Remarks Default value is 256.

MaxWidth Property

Applies To Resize

Description The maximum width of the form.

Setting Long integer.

Remarks MaxWidth is expressed in twips. 1 inch is 1440 twips.

Min Property

Applies To Common Dialog, Gauge, Progress Meter, Slider, Spin Button, Splitter

Description The minimum value of the control.

Setting Integer.

Remarks The Max and Min properties define the range of a control. Assigning a value higher than the Max value results in an error.

Font Common Dialog: The flag must be set to 8192 before using this property.

Print Common Dialog: The smallest number of pages allowed in the From field.

Gauge, Progress Meter, Slider, and Spin Button: The minimum number that the Value property can take.

Splitter: The minimum value (in twips) where the splitter can move.

MinHeight Property

Applies To Resize

Description The minimum height of the form.

Setting Long integer.

Remarks MinHeight is expressed in twips. 1 inch is 1440 twips.

Minute Property

Applies To CD Player

Description The current minute position (number of minutes into the current track).

Setting Integer.

Minutes Property

Applies To CD Player

Description The number of minutes in each CD track (excluding seconds).

Setting Array of integers.

Remarks This item is a zero based property array from 0 to TrackCount - 1.

This property is read-only. Assigning a value to this property causes an error.

This property returns the number of minutes for each track (no seconds). For example, if a track is 3 minutes and 12 seconds long, Minutes returns 3. To get the seconds for each track, use the Seconds property (page 141).

MinWidth Property

Applies To Resize

Description The minimum width of the form.

Setting Long integers

Remarks MinWidth is expressed in twips. 1 inch is 1440 twips.

NeedleStyle Property

Applies To Gauge

Description The visual style of the Gauge needle.

Setting The NeedleStyle property has the following settings:

Setting	Description	VBA
Line	Needle displays as a line.	TacmGauLine
Arrow	Needle displays as an arrow.	TacmGauArrow
Filled Arrow	Needle displays as a filled arrow.	TacmGauFilledArrow
Modern	Needle displays as a modern arrow.	TacmGauModern
Triangle	Needle displays as a triangle.	TacmGauTriangle

Remarks The thickness of the line is determined by the LineWidth property.

NoteBackColor Property

Applies To Notes

Description The popup edit window's BackColor.

Setting Long integer.

Remarks On the control's property sheet, select the color in the Color sheet.

In VBA, use a long integer to set this property.

NoteForeColor Property

Applies To Notes

Description The color for the text in the popup edit window.

Setting Long integer.

Remarks On the control's property sheet, select the color in the Color sheet.

In VBA, use a long integer to set this property.

NumLock Property

Applies To System Information

Description	The state of the [Num Lock] key.
Setting	Boolean. (True if NumLock is on, and false if it is off)
Remarks	This property is read-only. Assigning a value to this property causes an error.

OnAlarm Event

Applies To	Clock
Description	Fires when the AlarmTime is reached.
Syntax	Sub *ControlName*_OnAlarm ()
Remarks	This event does not fire if EnableAlarm is False. AlarmTime is based on the system time, not the clock time.

OnChange Event

Applies To	Notes, Slider, Spin Button, Tab
Description	Occurs when the user changes the value of the control.
Syntax	Sub *ControlName*_OnChange ()
Remarks	This event does not fire if the Value property is changed through code.

OnHighlight Event

Applies To	Icon Menu
Description	Occurs when an icon in the control is highlighted (when the mouse is on top of it).
Syntax	Sub *ControlName*_OnHighlight ()

OnNewPosition Event

Applies To	AVI, CD Player
Description	Fires when a new position is reached.
Syntax	Sub *ControlName*_OnNewPosition ()
Remarks	This event fires once every second for the CD Player Control, and once every millisecond for the AVI control.

OnNewTrack Event

Applies To	CD Player
Description	Fires when Play begins, and each time a new track on the CD is started.
Syntax	Sub *ControlName*_OnNewTrack ()

OnReplay Event

Applies To	AVI, Wave
Description	Occurs when the control replays a multimedia file.
Syntax	Sub *ControlName*_OnReplay ()
Remarks	This event is dependent on the RepeatCount and Repeat Properties.

OnSelect Event

Applies To	Icon Menu, Popup Menu
Description	Occurs when the user selects a menu item.
Syntax	Sub *ControlName*_OnSelect (item As Integer)

Argument	Description
Item	The item number of the zero based menu items.

OnSpin Event

Applies To Spin Button

Description Fires when a control button is clicked.

Syntax Sub *ControlName*_OnSpin (Increased As Integer)

Argument	Description
Increased	True (-1) if the up button was clicked, and False (0) if it was not.

Remarks This event fires after the OnChange event, and after Value is updated.

OnSplit Event

Applies To Splitter

Description Occurs when a splitter bar is moved.

Syntax Sub *ControlName*_OnSplit ()

Remarks This event is fired after the splitter bar is moved.

OnTimer Event

Applies To Timer

Description Occurs when the time specified by Interval has elapsed.

Syntax Sub *ControlName*_OnTimer ()

OnUnHighlight Event

Applies To Icon Menu

Description Occurs when an icon in the control loses the highlight.

Syntax Sub *ControlName*_OnUnHighlight ()

Remarks An item loses the highlight when the mouse is no longer on top of it.

Opened Property

Applies To Notes

Description The state of the Popup window.

Setting Boolean. (True if the popup editor is open, and false if it is not.)

Remarks This property is read-only. Assigning a value to this property causes an error.

Open Method

Applies To AVI, CD Player

Description Opens the resource.

Syntax *Integer* = *ControlName*.Open

Returns Boolean.

Remarks **CD Player**: Returns False if another program has the CD open, if there is no playable CD in the CD ROM drive, or if there is no CD ROM drive.

AVI control: Returns False if the FileName could not be opened.

Orientation Property

Applies To Icon Menu, Progress Meter, Slider, Spin Button, Splitter

Description The orientation of the control.

Setting The Orientation property has the following settings:

	Setting	Description	VBA
Icon Menu	Horizontal	Control is displayed horizontally.	TacmIcmHorizontal
	Vertical	Control is displayed vertically.	TacmIcmVertical
	Flowed	Control is displayed vertically and horizontally to maximize use of the display area.	TacmIcmFlowed
Progress Meter, Slider, Spin Button, Splitter			
	Horizontal	Control is displayed horizontally.	TacmOriHorizontal
	Vertical	Control is displayed vertically.	TacmOriVertical

Pause Method

Applies To AVI, CD Player, Wave

Description Pauses the control if it is playing.

Syntax *ControlName*.Pause

Returns Does not return a value.

Remarks This method differs from the Stop method in that it does not reset to the beginning of the file or CD.

Picture Property

Applies To About Box, Clock, Cursor, Enhanced Button, Gauge

Description Graphic to display in the control.

Setting The Picture property is only available at design time.

Remarks To load a picture at run-time, use the LoadPicture method.

Picture1 Property

Applies To Bitmap Effects

Description Graphic to display in the control.

Setting The Picture1 property is only available at design time.

Remarks To load a picture at run-time, use the LoadPicture1 method.

Picture2 Property

Applies To Bitmap Effects

Description Graphic to display in the control.

Setting The Picture2 property is only available at design time.

Remarks To load a picture at run-time, use the LoadPicture2 method.

PictureDown Property

Applies To Enhanced Button

Description Graphic to display in the control.

Setting The PictureDown property is only available at design time.

Remarks To load a picture at run-time, use the LoadPictureDown method.

PicturePosition Property

Applies To	Enhanced Button
Description	Location of the picture in the control.
Setting	The PicturePosition property has the following settings:

Setting	Description	VBA
Left	Picture appears to the left of the Caption text.	TacmEnbLeft
Top	Picture appears above the text.	TacmEnbTop
Right	Picture appears to the right of the text.	TacmEnbRight
Bottom	Picture appears below the text.	TacmEnbBottom

Play Method

Applies To AVI, CD Player, Wave

Description Plays the sound associated with the control.

Syntax *ControlName*.Play

Returns Does not return a value.

Remarks Play always starts at the current position in the file. When you instantiate the control, specify the file name and use Play to play the file from the beginning.
The Stop method resets to the beginning of the file. The Pause retains the current position.

Position Property

Applies To AVI

Description Use the Position property to read or set the current millisecond position in the video file.

Setting The Position property accepts long values.

PrinterDefault Property

Applies To Common Dialog

Description Whether the Print dialog selections update the system’s default printer settings.

Setting Boolean. (True to update the system default with the user’s selections, and false to keep the system default.)

Register Method

Applies To Registry Control

Description Registers the specified library.

Syntax *ControlName.*Register *Filename*

Part	Description
Filename	String specifying the file you want to register.

Returns Does not return a value.

Repeat Property

Applies To AVI, CD Player, Wave

Description Whether the control replays when it is finished.

Setting Boolean (True to replay the file when it’s finished, and false to stop when the file is finished.)

RepeatCount Property

Applies To AVI, Wave

Description The number of times to play the selected file.

Setting Integer greater than or equal to 0.

Remarks A value of 0 signifies infinite repeat.

Resize Method

Applies To Resize

Description Updates the size of all controls on a form.

Syntax *ControlName*.Resize

Returns This method is typically called in the resize event of a form.

ResizeFonts Property

Applies To Resize

Description Whether to resize fonts when the form is resized.

Setting Boolean. (True to resize fonts when the form is resized, and false to keep fonts at their original size.)

ReturnFSAncestors Property

Applies To Browse for Folder

Description Whether selection limited to file system ancestors.

Setting Boolean. (True to limit selection to file system ancestors, and false to allow folders that are not part of the file system.)

Remarks An ancestor is a subfolder that is beneath the root folder.

ReturnOnlyFSDirs Property

Applies To Browse for Folder

Description Whether selection is limited to file system directories.

Setting Boolean. (True to limit selection to file system directories, and false to allow folders that are not part of the file system.)

Rewind Method

Applies To AVI, Wave

Description Resets play to the beginning of the file.

Syntax *ControlName*.Rewind

Returns Does not return a value.

RollOver Property

Applies To Spin Button

Description Whether the value property “rolls over” when you increment above the maximum value or decrement below the minimum value.

Setting Boolean. (True to “roll over” to the minimum after the maximum is reached, and to the maximum after the minimum is reached. False to stop incrementing when the maximum is reached, and to stop decrementing when the minimum is reached.)

Second Property

Applies To CD Player

Description The number of seconds into the current track.

Setting Integer.

Seconds Property

Applies To CD Player

Description The number of seconds in each CD track (excluding minutes).

Setting Array of integers.

Remarks This item is a zero based property array from 0 to TrackCount - 1.
This property is read-only. Assigning a value to this property causes an error.
This property returns the number of seconds on each track after the minutes property is taken into account. For example, if a track is 3 minutes and 12 seconds long, 12 Seconds returns 12. This property should be used in conjunction with the Minutes property (see page 134).

SetFormFooter Method

Applies To Resize

Description Initializes the form footer sizing information.

Syntax *ControlName*.SetFormFooter *FormFooterName*

Returns Does not return a value.

Remarks This method initializes the resize ratios for the form footer.

SetFormHeader Method

Applies To Resize

Description Initializes the form header sizing information.

Syntax *ControlName*.SetFormHeader *FormHeaderName*

Returns Does not return a value.

Remarks This method initializes the resize ratios for the form header.

SetKeyValue Method

Applies To Registry Control

Description Sets the value of a key.

Syntax *ControlName*.SetKeyValue *RootKey*, *Key*, *Value*, *Data*

Part	Description
RootKey	Long integer that specifies the root key of the key to set.
Key	String that specifies the key to set.
Value	String specifying the key value.
Data	Variant that contains the value to set the key to.

Returns Does not return a value.
To set the default data for a key, specify a blank string (“”) as the value.

SetText Method

Applies To Clipboard

Description Copies text to the clipboard.

Syntax *ControlName*.SetText *Text*

Part	Description
Text	String specifying the text to copy to the clipboard.

Returns Does not return a value.

SetPicture Method

Applies To Clipboard

Description Copies an image to the clipboard.

Syntax *ControlName*.SetPicture(*Picture*)

Part	Description
Picture	Variant specifying the image data to copy into the clipboard.

Returns Does not return a value.

SetValue Method

Applies To INI Control

Description Sets a value in an INI file.

Syntax *String* = *ControlName*.SetValue *Section*, *Entry*, *Value*

Part	Description
Section	String that specifies the section heading in the initialization file.
Entry	String that specifies the entry whose value is retrieved.
Value	Integer that specified the new value of the key.

Returns Does not return a value.

Remarks This method uses the file specified in Filename.

ShadowColor Property

Applies To Border, Clock, Enhanced Button, Marquee, Spin Button, Text Effects

Description The color for the control’s shadow portion.

Setting Long integer.

Remarks On the control’s property sheet, select the color in the Color sheet.

In VBA, use a long integer to set this property.

Show Method

Applies To	About Box, Browse for Folder, Notes, Popup Menu
Description	Displays the window associated with the non-visual control.
Syntax	**About Box, Browse for Folder, and Notes:** *ControlName*.Show **Popup Menu:** *ControlName*.Show X, Y

Part	Description
X	X coordinate (in twips) for the current location of the mouse pointer.
Y	Y coordinate (in twips) for the current location of the mouse pointer.

Returns	Does not return a value.

ShowCalendar Property

Applies To	DateTime Picker
Description	Whether to allow users to use a drop-down calendar to select a date.
Setting	Boolean. (True to show a drop-down calendar, and false to show no calendar.)
Remarks	This property does not apply if the Format property is Time.

ShowColor Method

Applies To	Common Dialog
Description	Display the color selection common dialog box.
Syntax	*ControlName*.ShowColor

ShowDate Property

Applies To	Clock
Description	Whether to show the date on the digital clock.
Setting	Boolean. (True to show the date, and false to show time only.)

ShowFont Method

Applies To	Common Dialog
Description	Display the Font Common Dialog.
Syntax	*ControlName*.ShowFont
Returns	Does not return a value.

ShowHelp Method

Applies To	Common Dialog.
Description	Display Windows help (HLP) or HTML Help (CHM) file.
Syntax	*ControlName*.ShowHelp
Returns	Does not return a value.

ShowHourTicks Property

Applies To Clock

Description Whether to show hour tick marks on the clock.

Setting Boolean. (True to show hour tick marks, and false to hide them.)

Remarks This property only applies when the clock is analog.

ShowMinuteTicks Property

Applies To Clock

Description Whether to show minute tick marks on the clock.

Setting Boolean. (True to show minute tick marks, and false to hide them.)

Remarks This property only applies when the clock is analog.

ShowNone Property

Applies To DateTime Picker

Description Show a check box, which enables or disables typing in the DateTime Picker.

Setting Boolean. (True to show the "ShowNone" check box.)

Remarks When ShowNone is True, users can still use the Calendar drop-down to select a date. Set ShowCalendar to False to disable the calendar drop-down.

ShowOpen Method

Applies To Common Dialog

Description Display the Open Common Dialog.

Syntax *ControlName*.ShowOpen

Returns Does not return a value.

ShowPrinter Method

Applies To Common Dialog

Description Display the Printer Common Dialog.

Syntax *ControlName*.ShowPrinter

Returns Does not return a value.

ShowSave Method

Applies To Common Dialog

Description Display the Save Common Dialog.

Syntax *ControlName*.ShowSave

Returns Does not return a value.

ShowSeconds Property

Applies To Clock

Description Whether to show seconds on the clock face.

Setting Boolean. (True to show seconds, and false to hide them.)

SmallChange Property

Applies To	Slider, Spin Button
Description	The number of units the Value property changes when a small change event occurs.
Setting	Integer.
Remarks	The default for the SmallChange property is 1.

StartLocation Property

Applies To	Gauge Control
Description	The starting location of the gauge dial.
Setting	The StartLocation property has the following settings:

Setting	Description	VBA
Left	Dial starts on the left side of the control.	TacmGauLeft
Top	Dial starts at the top of the control.	TacmGauTop
Right	Dial starts on the right side of the control.	TacmGauRight
Bottom	Dial starts at the bottom of the control.	TacmGauBottom

StartPicture Property

Applies To	Bitmap Effects
Description	The picture (1 or 2) that is used as the starting picture in the Animate method.
Setting	The StartPicture property has the following settings:

Setting	Description	VBA
1	Start with the bitmap in Picture1.	TacmBmpPicture1
2	Start with the bitmap in Picture2.	TacmBmpPicture2

Stepping Property

Applies To	Marquee
Description	The size of a step in the marquee.
Setting	Integer.

Steps Property

Applies To	Bitmap Effects
Description	The number of steps in the animation.
Setting	Positive integer.

Stop Method

Applies To	AVI, CD Player, Wave
Description	Stops the control if it is playing.
Syntax	*ControlName*.Stop
Returns	Does not return a value. The Stop method stops playing the file and rewinds to the beginning. If you want to stop the control without rewinding, use the Pause method.

Stretch Property

Applies To	AVI

Description Whether the video is stretched to the size of the control.

Setting Boolean. (True to stretch the video display to the size of control, and false to display the video in its original size.)

SystemDir Property

Applies To System Information

Description The computer’s Windows System folder.

Setting String.

Remarks This property is read-only. Assigning a value to this property causes an error.

TabCount Property

Applies To Tab Control

Description The number of tab pages.

Setting Integer.

TabLocation Property

Applies To Tab Control

Description Location of the tabs on the control.

Setting The TabLocation property has the following settings:

Setting	Description	VBA
Left	Show tabs on the left of the control.	TacmTabLeft
Top	Show tabs at the top of the control.	TacmTabTop
Right	Show tabs on the right of the control.	TacmTabRight
Bottom	Show tabs at the bottom of the control.	TacmTabBottom

Remarks Use a True Type font if you are setting this property to anything other than Top.

TabText Property

Applies To Tab Control

Description Text to display on the tab.

Setting Array of strings.

Remarks This item is a zero based property array. Use an index number from 0 to TabCount - 1.

TempDir Property

Applies To System Information

Description The system’s temporary folder.

Setting String.

Remarks This property is read-only. Assigning a value to this property triggers an error.

TempFileName Property

Applies To System Information

Description Temporary file name.

Setting String.

Remarks This property is read-only. Assigning a value to this property triggers an error.
Every time this property is read, it returns a different temporary file name.

TextLocation Property

Applies To Notes, Text Effects

Description Location of the text in the control.

Setting The TextStyle property has the following settings:

	Setting	Description	VBA
Notes	Left	Show text on the left of the control.	TacmNteLeft
	Right	Show text at the top of the control.	TacmNteTop
	Top	Show text on the right of the control.	TacmNteRight
	Bottom	Show text at the bottom of the control.	TacmNteBottom
Text Effects	Center	Show text in the center of the control.	TacmTefCenter
	Left	Show text on the left of the control.	TacmTefLeft
	Right	Show text at the top of the control.	TacmTefTop
	Top	Show text on the right of the control.	TacmTefRight
	Bottom	Show text at the bottom of the control.	TacmTefBottom

Remarks **Notes**: Bottom is the default option.

Text Effects: Center is the default option.

TextStyle Property

Applies To Clock, Enhanced Button, Marquee, Text Effects

Description The display style for the control’s text.

Setting The TextStyle property has the following settings:

Setting	Description	VBA
Normal	No special formatting.	TacmTxtNormal
Three-D	Three dimensional effect.	TacmTxtThreeD
Sunken	Sunken effect.	TacmTxtSunken

ThemeEnable Property

Applies To Clock, Marquee

Description Whether the control respects the Windows Theme

Setting Boolean (True to respect the Windows Theme)

Remarks When ThemeEnable is True, certain appearance properties (such as BackColor) are not respected. Instead, the appearance is inherited form the Windows Theme.

TickFrequency Property

Applies To Gauge, Slider

Description The frequency of tick marks drawn on the control.

Setting Integer.

Remarks The number of ticks is calculated by subtracting Min from Max. The number of ticks drawn is calculated by dividing the number of ticks by TickFrequency.

E.g. if TickFrequancy = 2, every other tick mark is shown.

TickStyle Property

Applies To Gauge, Slider

Description The appearance of tick marks on the control.

Setting The TickStyle property has the following settings:

	Setting	Description	VBA
Gauge	Normal	Show Tick marks.	TacmGauTickNormal
	None	Don't show ticks marks.	TacmGauTickNone

	Setting	Description	VBA
Slider	Left/Bottom	Display tick marks on the bottom or left.	TacmSliLeftBottom
	Right/Top	Display tick marks on the top or right.	TacmSliRightTop
	Both	Display tick marks on both sides of the slider.	TacmSliBoth
	None	Don't display ticks marks.	TacmSliNone

Time Property

Applies To Clock

Description The time displayed in the control.

Setting Date.

Remarks The default value is the current date and time.

Title Property

Applies To Browse for Folder, Notes

Description **Browse for Folder:** The text displayed in the dialog box (for Browse for Folder), or in the title bar of the popup editor (for Notes).

Setting String.

ToPage Property

Applies To Common Dialog

Description The value for the To text box.

Setting Integer.

Remarks This property is valid only when the flag is set to 2.

Track Property

Applies To CD Player

Description The current Track position, or the number of tracks into the CD.

Setting Integer.

Remarks If you set this property to an invalid value, the value is ignored (no error is triggered).

Tracks Property

Applies To CD Player

Description The number of tracks on the CD.

Setting Integer.

Remarks This property is read-only. Assigning a value to this property causes an error.

Units Property

Applies To	Clock
Description	The number style of the digital display on the clock.
Setting	The Units property has the following settings:

Setting	Description	VBA
12 Hour	Display the time in 12-hour increments.	TacmClk12Hour
24 Hour	Display the time in 24-hour increments.	TacmClk24Hour

UserName Property

Applies To	System Information
Description	The Windows network user name.
Setting	String.
Remarks	The UserName property is read-only. Assigning a value to this property causes an error.

Validate Property

Applies To	Browse for Folder
Description	Whether to validate the selected path.
Setting	Boolean. (True to validate the path, and disable the [OK] button when the path is invalid. False to accept invalid paths.)
Remarks	A path is considered invalid if it does not exist.

Value Property

Applies To	Browse for Folder,
Description	The value of the control.
Setting	The Value settings depend on the control.

Control(s)	Setting
Browse for Folder	String containing the folder path.
DateTime Picker	Date containing the date/time in the control.
Enhanced Button	Boolean containing the toggle state of the button: 0 if toggled down, and -1 if toggled up. This property is only valid when ButtonStyle is Toggle.
Tab	Integer containing the number of the current tab. Value is 0-based (i.e. Tab 1 = Value 0).
Gauge, Progress Meter, Slider, and Spin	Integer between the control’s Min and Max properties.
Digital Display, Marquee, Notes, and Text Effects	String containing the control’s text.

Remarks The value property can be databound (see page 26).

VUColor(n) Property

Applies To Progress Meter

Description The color for the specified segment of the progress bar.

Setting Long integer.

Remarks If Style is set to VU Bar, the control divides the number progress bar into four sections, which are defined by VUPercent2, VUPercent3, and VUPercent4. The VUColor properties define the colors for each segment:

- **VUColor1** is the color of the first segment, from the beginning of the progress bar to VUPercent2.
- **VUColor2** is the color of the second segment, from VUPercent2 to VUPercent3.
- **VUColor3** is the color of the third segment, from VUPercent3 to VUPercent4.
- **VUColor4** is the color of the last segment, from the VUPercent3 to the end of the progress bar.

On the control's property sheet, select the color in the Color sheet.

In VBA, use a long integer to set this property.

VUPercent(n) Property

Applies To Progress Meter

Description The percentage where the specified VUColor property begins.

Setting Integer between 0 and 100.

Remarks If Style is set to VU Bar, the control divides the number progress bar into four sections, which are defined by VUPercent2, VUPercent3, and VUPercent4.

- **VUPercent2** is the percentage where VUColor1 ends, and VUColor2 begins.
- **VUPercent3** is the percentage where VUColor3 ends, and VUColor3 begins.
- **VUPercent4** is the percentage where VUColor3 ends, and VUColor4 begins.

WindowsDir Property

Applies To System Information

Description The system's Windows folder.

Setting String.

Remarks This property is read-only. Assigning a value to this property causes an error.

WindowsType Property

Applies To System Information

Description The type of Windows currently running.

Setting The WindowsType property has the following settings:

Description	VBA
Operating system is Windows 95.	TacmSysWin95
Operating system is Windows NT.	TacmSysWinNT
Operating system is neither Windows 95 nor Windows NT.	TacmSysOther

WindowsVersion Property

Applies To System Information

Description The version of Windows currently running.

Setting String

Remarks This property is read-only. Assigning a value to this property causes an error.

WordWrap Property

Applies To Marquee, Text Effects

Description Whether to wrap the text to a new line when it extends beyond the control's width.

Setting Boolean. (True to wrap text, and false to keep text on one line.)

Chapter 8: Product Support

This chapter provides information on troubleshooting problems that arise and obtaining support for Total Access Components.

Topics in this Chapter

- **Troubleshooting**
- **Web Site Support**
- **Technical Support Options**
- **Contacting Technical Support**

Troubleshooting

There are many resources available to resolve issues you may encounter. Please check the following:

Readme File

Check the README file for the latest product information. The README file is located in the folder where you installed the product.

Product Documentation

We’ve spent a great deal of care and time to make sure the Total Access Components manual and help file are very detailed. Check the Table of Contents and Index for your question, and read the appropriate pages.

Web Site Support

The FMS web site contains extensive resources to help you use our products better. Resources include product updates, frequently asked questions (FAQs), newsgroups, information on new versions, betas, and other resources.

Web Site

The FMS web site is located at:

```
www.fmsinc.com
```

News and important announcements are posted here.

Support Site

The main support page is located at:

```
www.fmsinc.com/support
```

From this page, you can quickly locate the other support resources.

Product Updates

FMS takes product quality very seriously. When bugs are reported and we can fix them, we make the updates available on our web site. If you are encountering problems with our product, make sure you are using the latest version.

Product updates can also be checked using the update wizard. See **Using the Update Wizard** on page 17 for details.

Frequently Asked Questions (FAQs)

Common questions and additional information beyond what is in the manual is often available from our FAQs.

Newsgroups

FMS also has general and product specific newsgroups. Connect with FMS Technical Support and other users there. Share your experiences, learn from others, and ask your questions in our virtual community:

```
www.fmsinc.com/support/newsgrp.htm
```

Or visit our web site for additional information.

Microsoft Patches

Our support site also includes links to Microsoft patches that are related to our products. Make sure you're using the latest versions by checking here or visiting the Microsoft site.

Technical Support Options

FMS is committed to providing professional support for all of our products. We offer free access to our online FAQs and newsgroups. Bug reports, feature requests, suggestions, and general pre-sales questions related to our products are always available at no cost.

Additional maintenance plans are available to provide subscribers with enhanced technical support. This is the best way for you to stay current with the rapidly changing technologies that impact project development, and to ensure you are getting the maximum return from your software investment.

Please visit our web site, www.fmsinc.com, for the most up-to-date information.

Features & Benefits	Premium	Incident	Standard
Access to FAQs	✔	✔	✔
Access to Newsgroups	✔	✔	✔
Minor Upgrades/Bug Fixes	✔	✔	✔
Telephone Support	✔	Per incident	First 30 Days
Email Support	✔	Per incident	First 30 Days
Priority Response Time [1]	✔	✔	
Senior Engineer Support Team	✔	✔	
Email Project for Testing	✔	✔	
Programmatic Code Assistance [2]	✔	✔	
Major Upgrades for Current Version (not between Access versions)	✔	Additional fee	Additional fee
Cost	Annual Fee	Per Incident	Included

1. Response generally within two business days. Actual resolution may take longer depending on complexity of the issue reported.
2. Custom Programming implementation is not provided in our Support Maintenance plans. For products that include a programmatic interface, we can provide instructions for using our programmatic interface, and show examples, but we do not implement this into your projects. This service is available from our Professional Solutions Group.

Premium Subscription

The Premium Subscription is the ideal option for customers seeking the highest level of support from FMS. The annual fee entitles you to telephone and email technical support from a senior support engineer.

From time to time, FMS may release new versions of existing products which add new features. These are point releases (e.g. from version 14.0 to 14.1) and are different from new builds that correct problems in existing features (e.g. from version 14.00.0001 to 14.00.0002).

These point releases are available for a nominal upgrade fee to existing customers. Premium Technical Support subscribers receive these upgrades automatically and for no additional charge during their subscription term.

NOTE: Upgrades between versions (for instance going from Access 2007 to Access 2010) are not considered Point Release Upgrades and are not included in the Premium Subscription.

Subscriptions are available for a twelve month period, and may be purchased at any time. You must be the registered owner of the product to purchase a subscription and the only person contacting FMS for support under the subscription.

Please ensure you have purchased the Subscription you need for Total Access Components.

Per Incident

Our Per Incident package is available individually or by purchasing multiple incidents in advance. The Per Incident support package provides telephone and email technical support from a Senior Technical Support Engineer for resolving one incident.

An incident is defined as a single question related to one of our products. The Per Incident period is from start to finish (report of the incident to resolution) for a single incident. If you anticipate multiple questions for a single product, we recommend purchasing the Premium Subscription.

Standard Subscription

Our Standard Subscription comes with every product purchased for no additional cost. The standard subscription comes with access to our FAQs and newsgroups, and responses to bug reports and feature requests for that version.

Please note that the person requesting support must also be the registered user of the product. Registration is required and will be requested by our Technical Support professionals.

Contacting Technical Support

If the troubleshooting suggestions and other support resources fail to resolve your problem, please contact our technical support department. We are very interested in making sure you are satisfied with our product.

Registering Your Software

Please register your copy of Total Access Components at:

```
www.fmsinc.com/support
```

You must be registered to receive technical support. Registration also entitles you to free product updates, notifications, information about upcoming products, and beta invitations. You can even receive free email notification of our latest news.

Support for Licensed Developers Only

FMS provides technical support for licensed developers using or distributing Total Access Components. FMS does not, however, provide technical support or customer service for users of your applications that use our controls—this is your responsibility.

Contact Us

The best way to contact us is via email at:

```
Support@fmsinc.com
```

Please provide detailed information about the problem that you are encountering. This should include the name and version of the product, your operating system, and the specific problem. If the product generated an error file, please submit that as well.

With email, technical support issues can be more accurately resolved and tracked in our internal technical support system. Email also gives us more time to understand the entire problem and allows our technical support staff to contact the developers with the entire story when necessary. Please bear in mind that a unique issue may involve meetings between the technical support staff and product developers, so your patience is appreciated.

Microsoft Technical Support

FMS only provides technical support for its products. If you have questions regarding Microsoft products, please contact Microsoft technical support.

Index