Microsoft Office 97 Programmability Features

Contents

- Introduction
- Improved Programming Capability
- New and Improved Object Models
- Tools for Building Connected Solutions
- Microsoft Office 97 Applications Programmability Enhancements

Introduction

This document is a guide to the new and enhanced programmability features in the Microsoft® Office 97 suite.

Microsoft Office is not only the best set of productivity applications for all types of users, but also a development platform for building custom business solutions. Today, more than one million developers are creating solutions based on Microsoft Office applications.

Microsoft Office 97 enhances and extends the features that make it possible for developers to build custom solutions with Microsoft Office. Microsoft Office 97 improves on the development foundation strategy first introduced in Office 4. *x* and continued with Office for Windows® 95 in three key areas:

- Improved programming capability. A new version of the Visual Basic® programming system, Applications Edition version 5.0, is now in Microsoft Excel, Microsoft Word, the Microsoft PowerPoint® presentation graphics program, and Microsoft Access. The integrated development environment (IDE) of Visual Basic for Applications has been completely redesigned and standardized for greater consistency across host applications. In addition, IntelliSense technology is now available to the developer, making writing code in Office 97 easier than ever. ActiveX™ Controls support in Office documents and Microsoft Forms enable developers to add rich interactive capabilities to their solutions easily.

- New and improved object models. Office 97 provides new applications and functionality that are exposed to the developer in more than 550 objects. Office 97 also

PENGAD-Bayonne, N. J. GOVERNMENT EXHIBIT 2214

developer in more than 350 objects. Office 97 also introduces new shared components such as OfficeArt, the Office Assistant, command bars, and FindFile that developers can make optimal use of for their own custom solutions.

- Tools for building connected solutions. Microsoft Office 97 incorporates and exposes several new technologies to take more effective advantage of the connected environment. Features such as ODBCDirect, Jet 3.5, HTML support, and hyperlinks increase the power and flexibility in accessing corporate data and provide seamless integration to both Intranet and Internet environments.

Top

Improved Programming Capability

Visual Basic for Applications is a shared development environment that enables developers for Microsoft Office to create custom business solutions. Visual Basic for Applications includes a language engine, a powerful editor, an Object Browser, and debugging tools. Microsoft Office 97 provides developers with more depth and flexibility in building custom solutions with the extension of Visual Basic for Applications and improvements to the development environment.

Visual Basic for Applications, Version 5.0

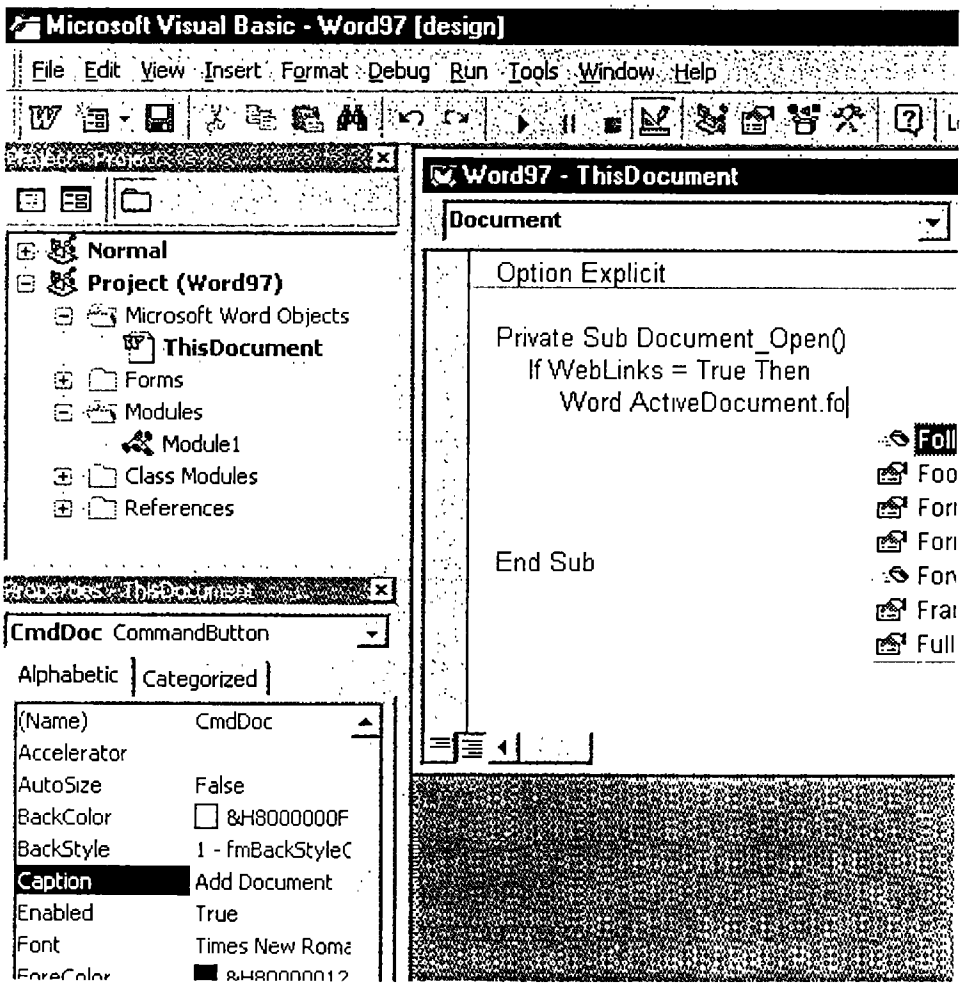
Visual Basic for Applications is now included in Microsoft Word and Microsoft PowerPoint. This inclusion makes it much easier for developers to create custom solutions using each Microsoft Office product and to reuse code across the applications. Moreover, with the recent licensing of Visual Basic for Applications to third parties, developers will be able to use standard tools to capture the functionality of Microsoft Office and third-party software products to build vertical business solutions that share a high degree of consistency for users.

Integrated Development Environment

The integrated development environment of Visual Basic for Applications is an entirely new environment for developers for Microsoft Office. The most significant change is that the new Visual Basic Editor exists outside the host application window. This enables developers to write code in Visual Basic for Applications and simultaneously view the results of their programming in the host Office application. Although the IDE exists outside the application window, it runs in the same memory space as its host, thereby benefiting from tight integration for event handling, as well as enhanced performance. The IDE also provides the following:

- Enhanced editor with syntax checking, color-coded syntax, and support for code drag-and-drop

- New Project Explorer for navigating the project components
- Enhanced Properties window for setting and viewing object properties
- New debugging tools to help track program execution and find bugs
- Enhanced Object Browser for browsing and searching for properties and methods across object model libraries
- New IntelliSense features for instant syntax reference and object model assistance to reduce programming time
- Improved code security with improved password protection and encryption to lock both documents and projects (developers for Microsoft Access can remove source code from their projects for added protection)
- Conditional compilation for incorporating debugging code that will not execute in the final version of the application



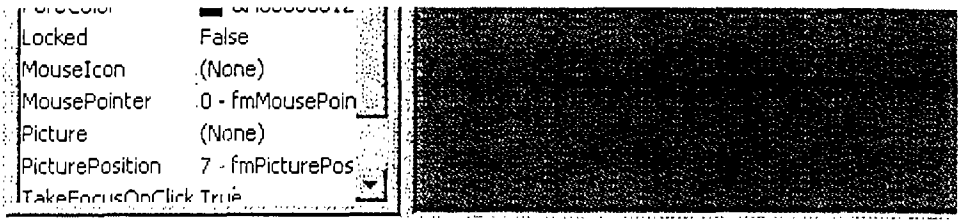


Fig. 1. Visual Basic Editor

Code Window

The IDE of Visual Basic for Applications includes a code window for the document, for any document sections that support code behind them (for example, Microsoft Excel sheets), and for each code and class module and form in the project.

Drag-and-drop functionality has been implemented throughout the editing environment. Developers can drag and drop code and variables between code windows, into the Watches window, into the Locals window, and across projects.

Project Explorer

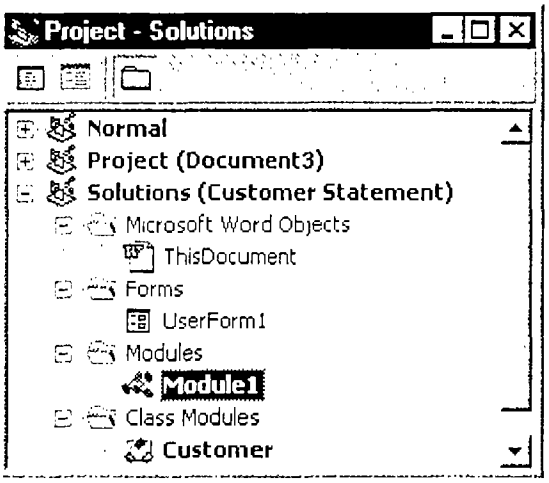


Fig. 2. Project Explorer

The Project Explorer displays the project components (specifically, forms, modules, and references) associated with each open project in an outline view. Each project appears as a new root in the outline control, enabling easy switching among different documents' projects for developers working on multiple projects simultaneously. A project exists for each open document and template.

The following project components are shown in the tree view:

- The document that owns the project (Word document, Microsoft Excel workbook, or PowerPoint presentation)

- Sections internal to the document that can have code behind them (for example, worksheets in Microsoft Excel)
- Forms that belong to the project
- Code and class modules that belong to the project
- References to other Microsoft Office documents (references are permitted to documents within the same application only)

Properties Window

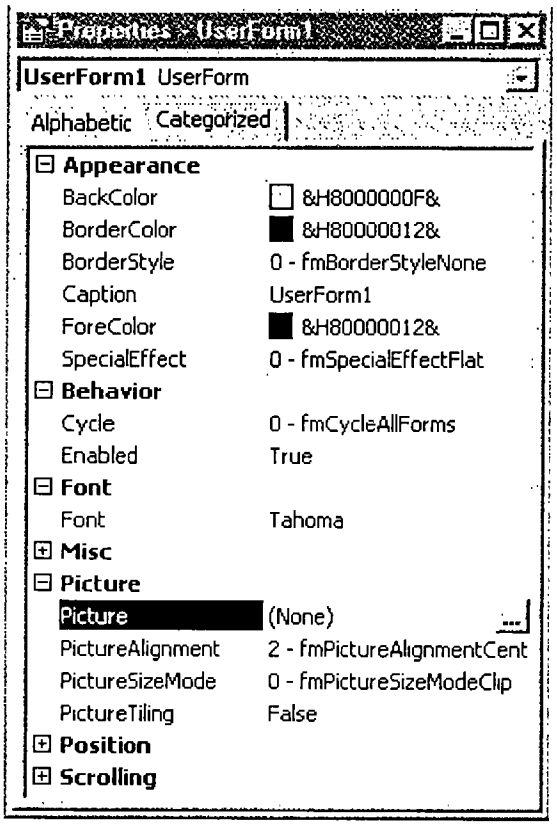


Fig. 3. Properties window

The Visual Basic Editor has a Properties window that displays the properties of Microsoft Office documents, forms, and controls. The Properties window can also be shown in the host application's window. For example, Word uses the Properties window to display properties of ActiveX Controls that are embedded in a Word document.

The Properties window has two tabs: Alphabetic and Categorized. The Alphabetic tab view provides an alphabetical list of properties. In the Categorized tab view, properties are grouped by category (for example, all properties related to color, font, or position). These categories can be expanded or collapsed in the Properties window.

Debugging Tools

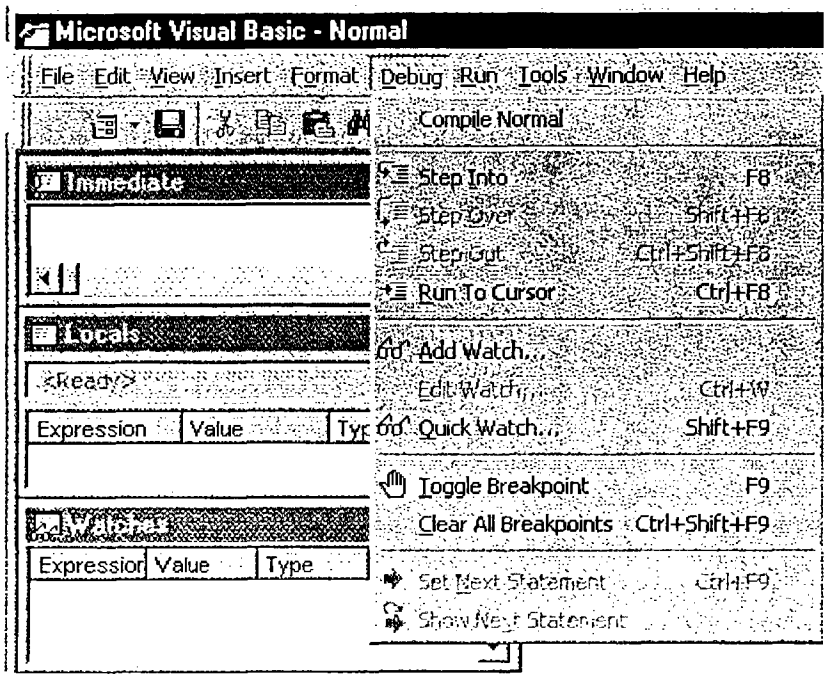


Fig. 4. Debug menu

Visual Basic for Applications includes new and enhanced debugging tools to help the developer identify compile errors, program logic errors, and run-time errors. The debugging tools in Visual Basic for Applications include a Locals window, Watches window, and an Immediate window. The Locals window also includes a Call Stack Browser that shows the current variable and enables the developer to jump to procedure definitions and references.

- Locals variables window automatically displays all the declared variables in the current procedure and their values.
- Watches window enables monitoring the value of a particular variable or expression. Code execution may be interrupted when a watch expression's value changes or equals a specified value.
- Immediate window instantly evaluates any expression or statement in Visual Basic, such as call to a Sub of Function.
- Call Stack displays a list of active procedure calls during break mode.
- To speed the debugging process, code can be dragged and dropped from the editor into the Immediate and Locals windows.

Object Browser

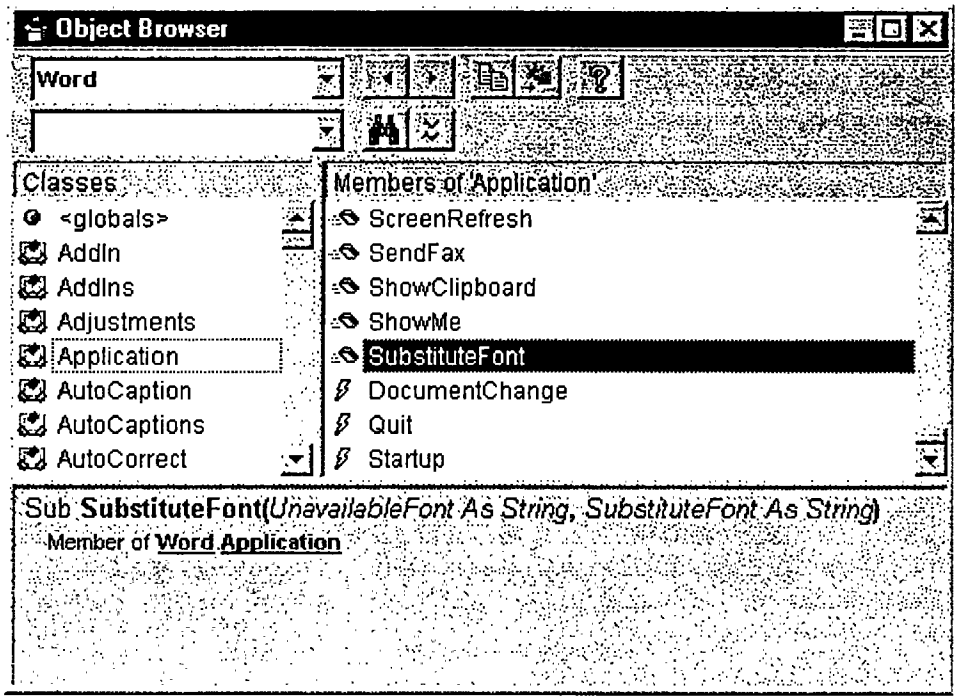


Fig. 5. Object Browser

The Visual Basic Editor has an improved Object Browser. The browser differentiates between built-in properties, custom properties, methods, event handlers, and user-defined procedures, as well as indicates globally accessible members. The browser also shows function return types, parameter names and types, and user-defined types and constants. Hyperlink jumps to referenced objects enable easy navigation of the object hierarchy. New to the Object Browser is the ability to search for objects and members across type libraries.

IntelliSense Features

Visual Basic for Applications, Version 5.0, brings IntelliSense technology to the developer, providing on-the-fly syntax and programming assistance and reference. The developer can choose to turn these automated features off and access them on demand through the Visual Basic for Applications menu or with keystroke combinations. The following features are available while in the code window, as well as from the Immediate window:

- **Complete Word**. Completes the word that is being typed, once enough letters are entered to make it distinct. (Keystroke equivalent: Ctrl+Alt+A.)

- **Quick Info**. When a procedure or method name is entered (followed by a space or an opening parenthesis), a tip automatically appears under the line of code writing. The tip gives syntax information about the procedure. (Keystroke

equivalent: Ctrl+I.)

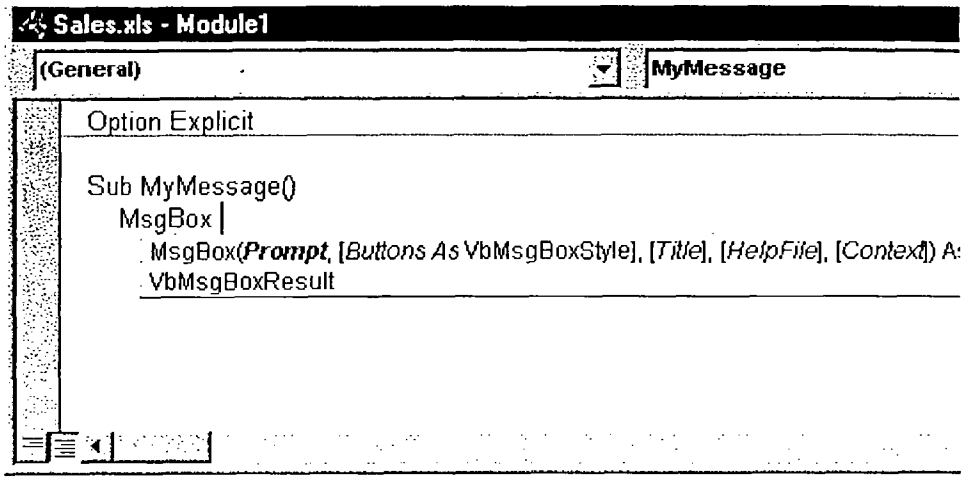


Fig. 6. Quick Info

- **List Properties/Methods**. Displays a pop-up menu listing the properties and methods available for the object that precedes the period. (Keystroke equivalent: Ctrl+J.)

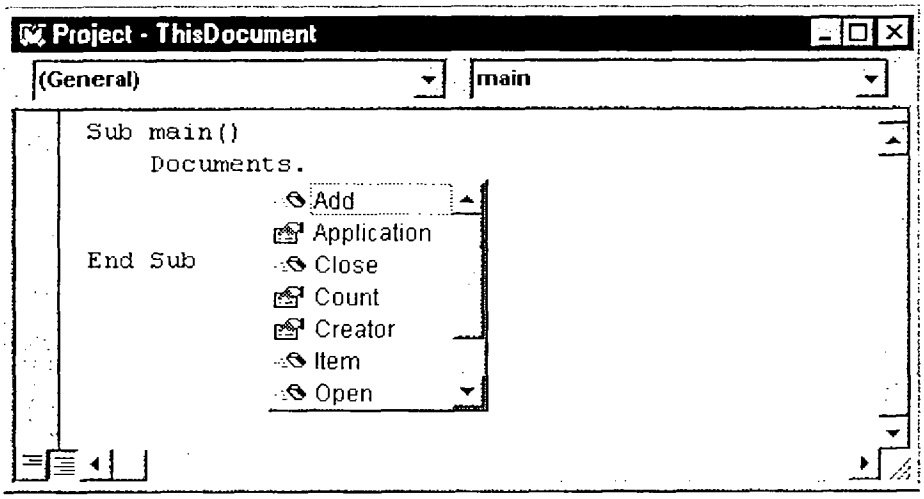


Fig. 7. List Properties/Methods

- **List Constants**. Displays a pop-up menu listing the constants that are valid choices for the property typed and that precede the equal sign (=). (Keystroke equivalent: Ctrl+Shift+J.)

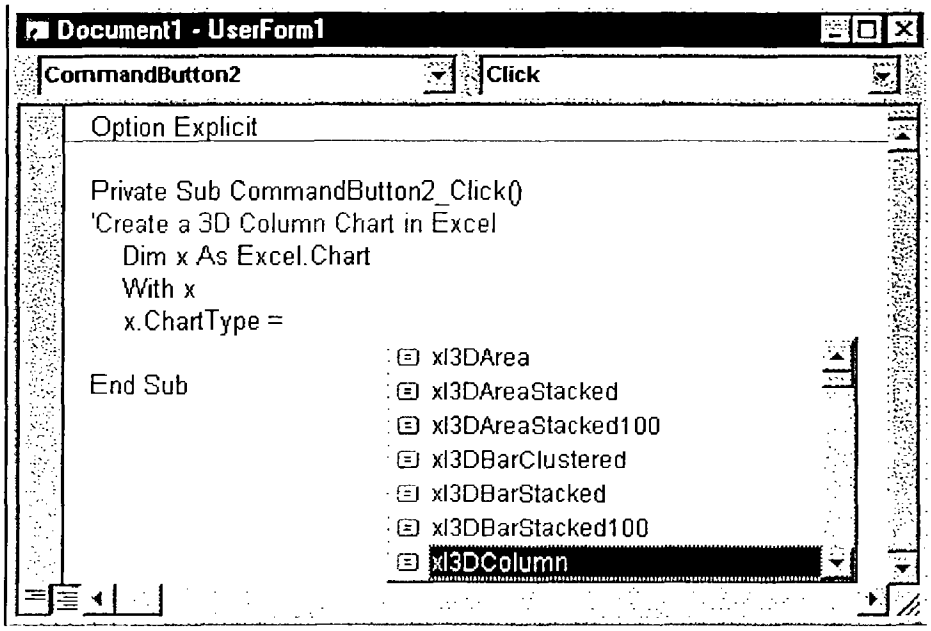


Fig. 8. List Constants

- **DataTips™ pop-up information**. When Visual Basic for Applications is in break mode and the cursor is placed over a variable, the value of the variable is displayed in a tooltips-like window.

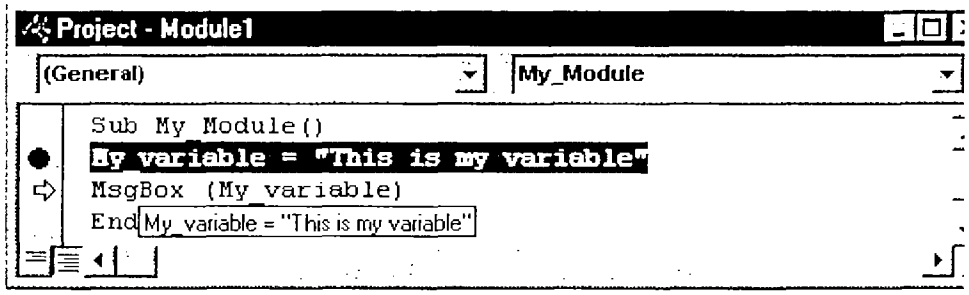


Fig. 9. DataTips

- **Margin Indicators**. Developers can set a breakpoint, set the next statement, or set a bookmark by clicking in the margin of the code editor.

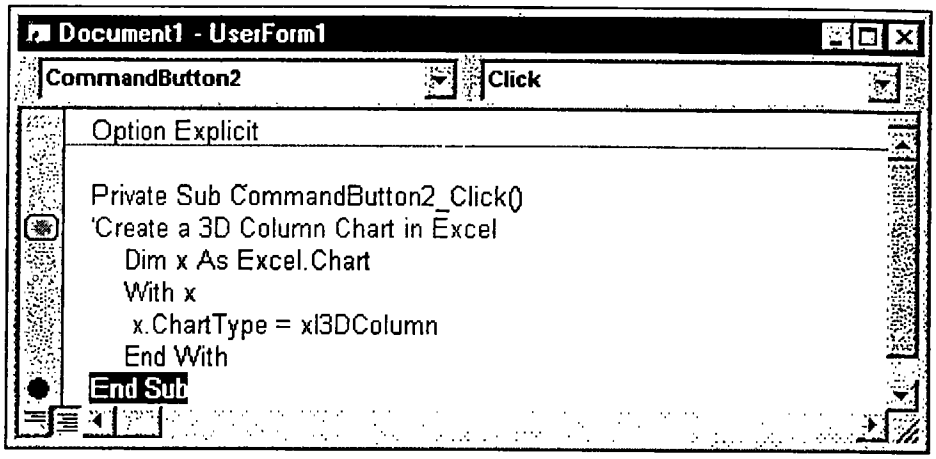


Fig. 10. Margin Indicators

Project Import and Export

Projects and their components (forms, class modules, and procedures) can be exported and imported for code storage and sharing. Exported project components are stored as standard text.

Block Comment and Uncomment

An Edit toolbar provides quick access to the Block Comment and Uncomment feature, which enables developers to select blocks of code and, with the click of a button, comment out the entire selected block of code.

Project Security

Developers need the ability to distribute their solutions in a secure and protected form to prevent modification to their code base. Projects associated with a Microsoft Office document can be protected without restricting access to the rest of the document. This protection is achieved by setting a protection property on the document in the property sheet. Developers using Microsoft Access can also remove the source code from databases that they distribute to others, thereby reducing the size of the database and protecting their intellectual property. For more information, see "Remove Source Code" in the Microsoft Access 97 section, later in this document.

Conditional Compilation

Developers can set compilation flags within their code to control the resulting behavior of the application. For example, developers may wish to include debugging code or custom messages during the development phase that they would not want end users to see. Conditional compilation enables developers to create multiple compilations of their application easily by setting flags in their

code.

Class Modules

Visual Basic for Applications now includes support for class modules. A class module can be thought of as a template for a user-defined object. Within a class module, developers can add public procedures that define custom methods and properties for the object. When an instance of the class is created, developers can apply these custom methods and properties as they would for any object. Modules can be shared across projects through drag-and-drop or export and import functionality or from within a Visual Basic-based automation server.

ActiveX Controls

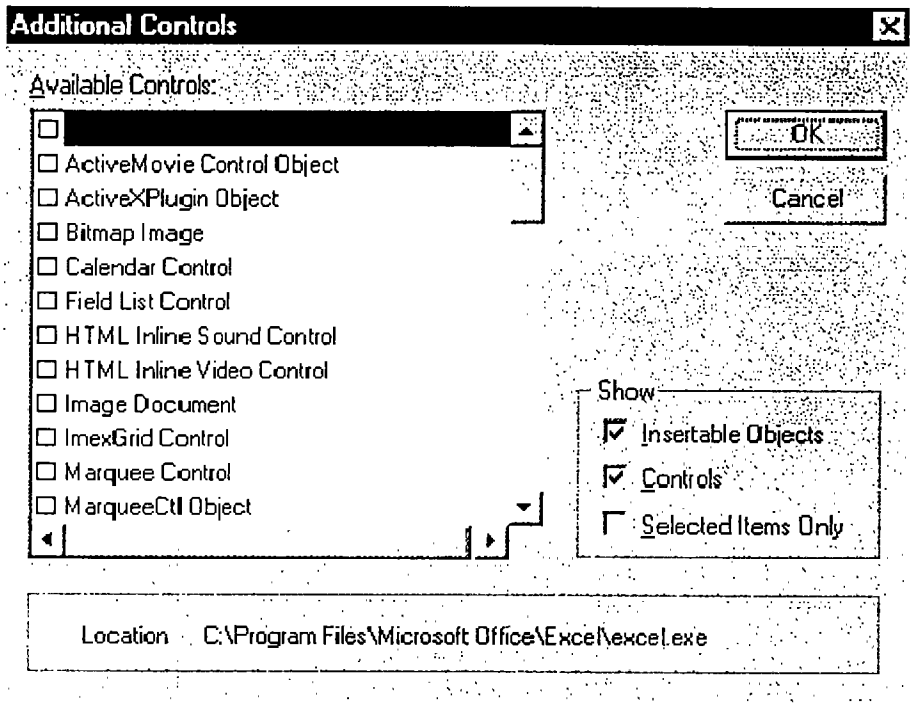


Fig. 11. ActiveX Controls menu

All of the Microsoft Office applications support ActiveX Controls, formerly called OLE Controls or custom controls. ActiveX Controls are prebuilt, reusable software components that enable developers to add rich interactive capabilities easily to their Microsoft Office solutions. In Microsoft Office 97, ActiveX Controls can be placed on Microsoft Forms or directly on Microsoft Office documents. More than 1,000 ActiveX controls are available today from third-party vendors.

Microsoft Forms

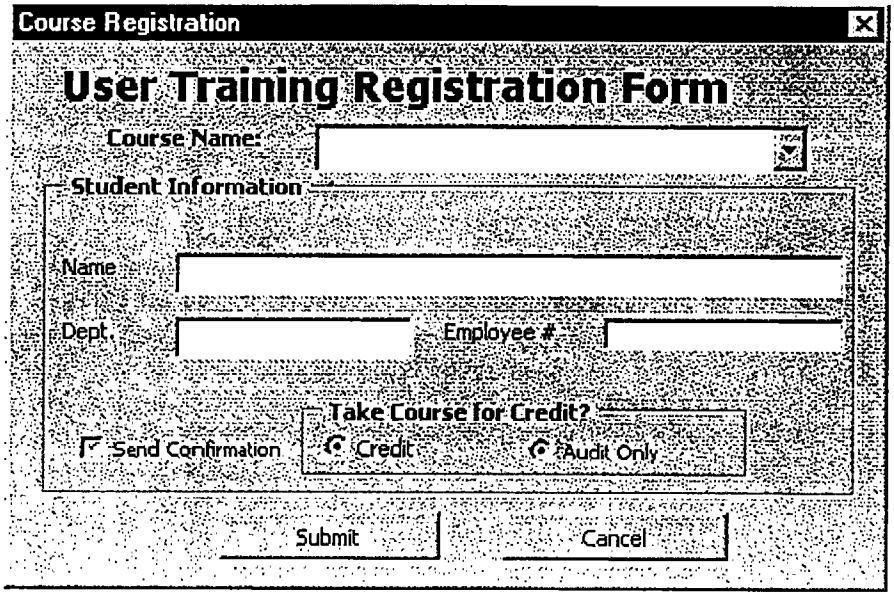


Fig. 12. Microsoft Forms

Microsoft Excel, Word, and PowerPoint share a powerful new Forms Designer, Microsoft Forms, which enables the developer to create custom dialog boxes. Because these applications use the same Forms Designer, developers learn how to create custom dialog boxes only one way for all three applications, and custom dialog boxes designed in one application can be shared with another.

Forms support the architecture for the next generation of ActiveX Controls and provide the following benefits:

- Superior performance (for example, time to save, load, redraw) when compared to today's windowed controls.
- Advanced features for forms design, including alignment and layout tools, drag-and-drop of controls, and in-place editing of captions.
- Ability to place the same controls on Microsoft Office documents.
- A broad set of controls, including all the standard interface controls and container controls such as frame and multipage controls.

Toolbox

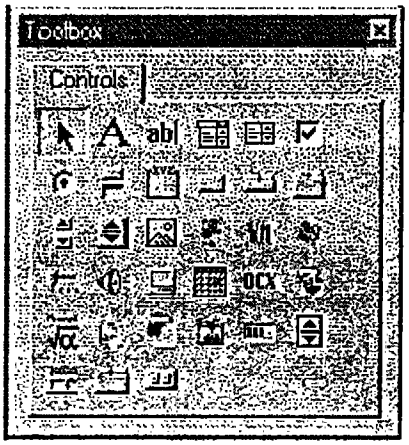


Fig. 13. Toolbox

The toolbox can be customized to display all registered controls in the system. Developers can fully customize the Microsoft Forms toolbox in the following ways:

- **Adding and removing pages**. Developers can customize and organize ActiveX Controls by adding pages to the toolbox.
- **Exporting and importing pages**. Developers can customize toolbox sharing and distribution of toolboxes and toolbox pages.
- **Control templates**. Developers can create control templates by dragging a control (or a number of "grouped" controls) from the form back to the toolbox. This action creates a template of that control, which can then be reused just like any other control. Templates carry the property settings of the parent control.
- **Customizing controls**. Developers can customize the icon representing the control or control templates created. The ToolTip presented for the control is also entirely customizable.

A second toolbox is used to place ActiveX Controls directly on Microsoft Office documents. Developers can maintain separate toolboxes for managing controls for forms and controls for documents.

△Top

New and Improved Object Models

At the center of the Microsoft Office development platform is the functionality in each Microsoft Office product that is available to

developers through ActiveX Automation. Features such as charting in Microsoft Excel or the database features in Microsoft Access are available to developers as building blocks to create custom solutions. Microsoft Office provides developers 100-percent coverage of the feature sets in each application, enabling them to create unique solutions in significantly less time. Also, Microsoft Office 97 provides more than 550 objects for reuse to the developer.

The developer can reuse the discrete pieces of functionality built into a Microsoft Office application through the application object model. An object model is a conceptual map or representation of an application's functionality depicting the hierarchy of the objects exposed by an application. Microsoft strives to provide developers consistency with object models to simplify the process of accessing components.

New components, such as the Office Assistant, OfficeArt, and command bars, share both code and their overall structure within each application. This consistency enables developers to work easily in different Microsoft Office applications, as well as to share code among the Microsoft Office applications. In Microsoft Office 97, more than 50 percent of the objects are shared, resulting in increased code reusability and improved product performance.

New Object Models

New functionality is introduced in Microsoft Office 97 that benefits end users by providing more features, and benefits developers by providing more objects to use to their greatest advantage. For example, the Microsoft Outlook™ desktop information manager provides a new set of full-featured objects that developers can use to provide scheduling and contact-management functionality. The Visual Basic for Applications, version 5.0, IDE and Microsoft Forms also expose their functionality through object models, enabling customization of the editor and forms environment. New object models are also provided for OfficeArt, The Office Assistant, Office command bars, and FileSearch.

OfficeArt

OfficeArt is the feature name for the new drawing and graphic tools in Word, Microsoft Excel, and PowerPoint. The new tools replace the existing drawing toolbars in each of these applications, and the code base is shared so the user interface is identical across the three applications. Additionally, Microsoft Graph accesses OfficeArt's fill dialog boxes and can apply OfficeArt fill effects to objects in graphs.

The new drawing and graphic tools make all the higher-end features of PowerPoint 95, such as gradient fills, available across Office 97. However, Office 97 also features rich new capabilities such as built-in WordArt, 3-D effects, and more than 100 AutoShapes.

The object model for OfficeArt is fully integrated into Word, Microsoft Excel, and PowerPoint, just as the OfficeArt tools are integrated in the user interface of the applications. Unlike with other Microsoft Office components described in this section (such as Office command bars and the Office Assistant), however, each application contains the OfficeArt object model with its own unique integration features. A core 90 percent of the model is identical across the three applications, and approximately 10 percent of the model is composed of properties, methods, and subobjects that are specific to each application.

Microsoft Office Assistant

Microsoft Office 97 introduces a new type of Help component -- the Office Assistant. The Office Assistant, the Assistant's balloon, and all the items inside the balloon are controlled programmatically through the Assistant portion of the Microsoft Office object model. Each of the following items are represented in Visual Basic through the Assistant object.

- Assistant actions
- Balloons
- Balloon headings, text, check boxes, buttons and labels

The following illustration shows the Office Assistant, the balloon, and all the programmable objects inside the balloon.

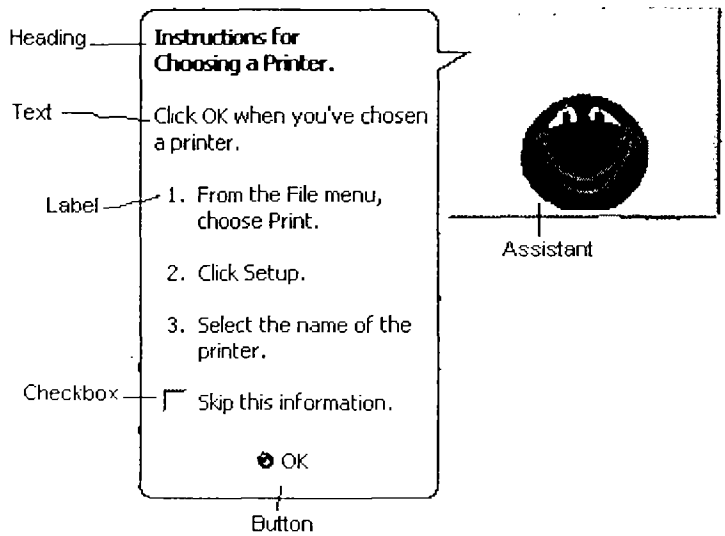


Fig. 14. Microsoft Office Assistant

Developers can use the Office Assistant in solutions to get a user's

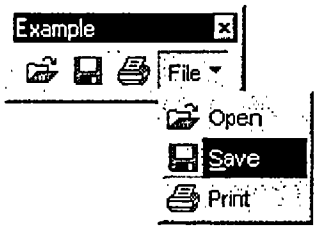
Developers can use the Office Assistant in solutions to get a user's attention, customize the Assistant's balloons to offer help in specific areas, or use the controls on the balloons to run Visual Basic for Applications procedures.

Microsoft Office Command Bars

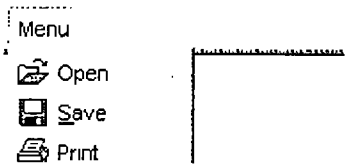
In Microsoft Office 97, toolbars, menu bars, and shortcut menus are all controlled programmatically through one type of object -- command bars. Developers can modify any built-in menu bar or toolbar and create and modify custom toolbars, menu bars, and shortcut menus. Because toolbars and menus are included in command bars, developers use the same kind of controls on each one. For example, the docked toolbar shown in the following illustration contains three buttons.

The menu shown in the following illustration contains the same three commands displayed as menu items.

In Microsoft Office 97, menu bars and toolbars can contain menus. The floating toolbar shown in the following illustration contains three buttons and a menu with the same three buttons displayed as menu items.



In Visual Basic for Applications, buttons and menu items are represented by CommandBarButton objects. The pop-up controls that display menus and submenus are represented by CommandBarPopup objects. In the following illustration, the control named Menu and the control named Submenu both are pop-up controls that display a menu and submenu, respectively. The menu and submenu both are unique command bar objects with their own set of controls.



In Microsoft Office 97, edit boxes, drop-down list boxes, and combo boxes can be added to any command bar. These three types of controls are all represented by CommandBar ComboBox objects.

FindFile and FileSearch

A new Application.FileSearch object has been added to provide access to all the searching power of the File Open and Advanced Find capabilities of Office applications. Like the File Open dialog box, FileSearch uses Find Fast indexes, when available, to make searching much faster. With FileSearch, developers can create customized search forms that take advantage of the powerful indexes and the search engine available in all Microsoft Office applications.

Top

Tools for Building Connected Solutions

At present, custom solutions integrate seamlessly with MAPI, ODBC, and LAN-based technologies to support distribution and access of enterprise data. However, Microsoft Office 97 incorporates and exposes several new technologies that allow developers to take more effective advantage of the connected environment. New features and technologies in Microsoft Office 97 provide seamless integration to intranet and Internet environments and increase power and flexibility in accessing corporate data.

Microsoft Office Web Solution

The new Web features in Microsoft Office 97 (including full hyperlink and searching capabilities) are available through the object model library. Developers can create powerful solutions that incorporate all the benefits of the Internet with the safety of a closed network. For example, with a custom-built intranet solution, an IS manager can define the Web search range on a server and control what files may be retrieved and the users who may retrieve them.

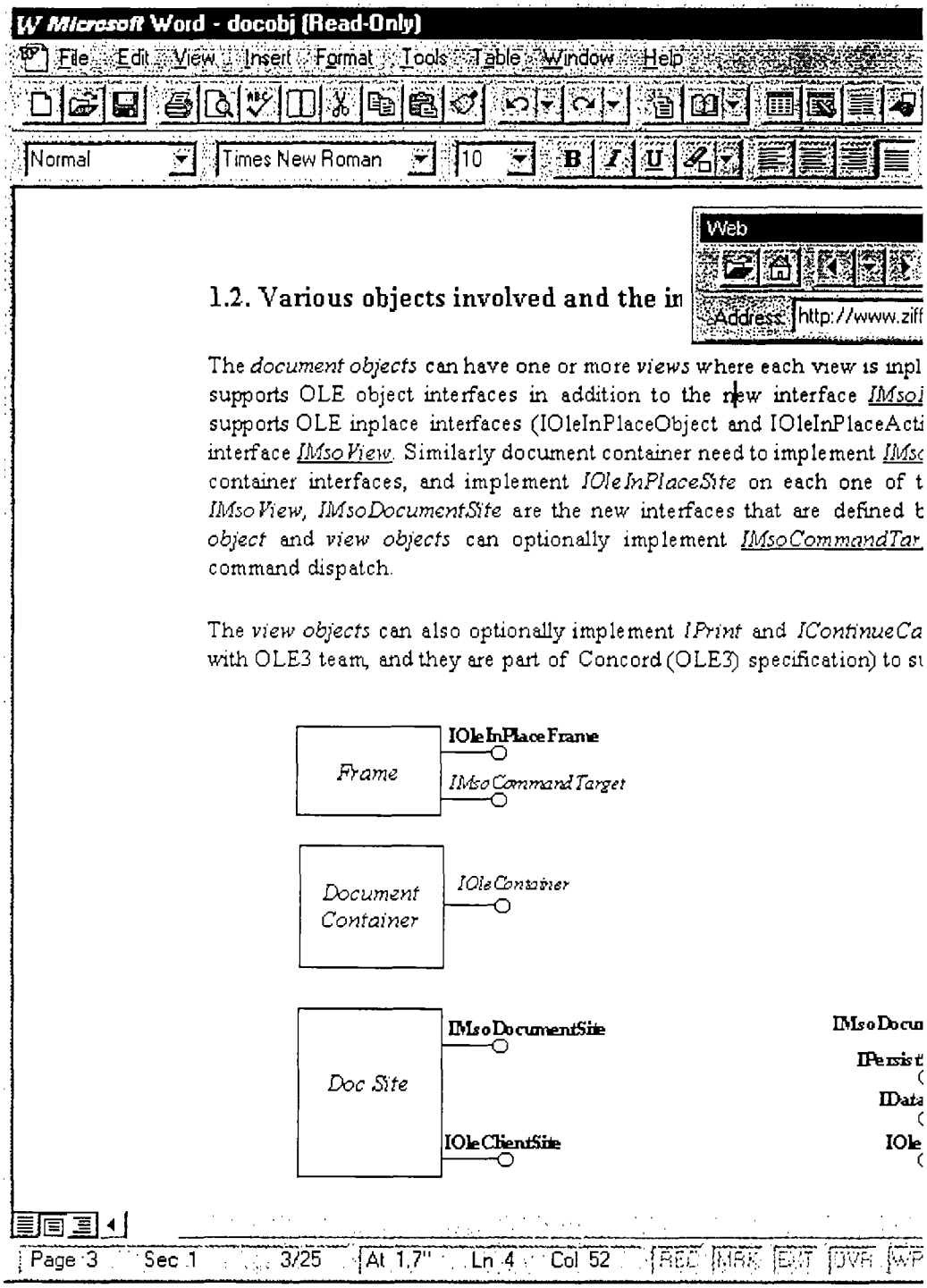


Fig. 15. Office Web

The Web features in Microsoft Office 97 are integrated consistently within all the application object models. Office-based Web

documents are ordinary Microsoft Office documents -- Word, Microsoft Excel and, PowerPoint -- not HTML documents.

Data Access Objects and Microsoft Jet

Data Access Objects (DAO) is the object model through which developers create data access solutions with the Microsoft Jet database engine. In Microsoft Office 97, a new version of DAO supports the following features in the new Jet 3.5 database engine:

- **Improved performance**. Rushmore™ query technology improvements and better cache handling means performance improvements for Jet databases.
- **Full and partial replication**. Partial replication enables developers to create a replica that contains only a subset of records in the full replica. This dramatically speeds up data exchange because only the necessary data is replicated, rather than the entire database. DAO also supports replication over the Internet and intranets.
- **Multiuser improvements**. Jet 3.5 provides improved methods of handling locking to provide optimal performance in multiuser scenarios.
- **Recordset control**. Developers now have programmatic control over the number of records returned from a SQL pass-through query.
- **Reduced database growth**. Empty space in the database is reclaimed significantly faster in Jet databases, reducing the growth rate of the database during application development.

ODBCDirect

DAO 3.5 introduces a new client/server connection mode called ODBCDirect. ODBCDirect establishes a connection directly to an ODBC data source without loading the Microsoft Jet database engine into memory. ODBCDirect is useful in situations that require specific features of ODBC. ODBCDirect provides fast access to remote data (on ODBC-compliant databases) and provides support for the following:

- Asynchronous processing
- Parameter passing
- Limiting number of rows returned
- Direct access to ODBC

- Batch updating

Universal Client

To help companies take advantage of Microsoft Office as a universal client to enterprise systems, Microsoft Office supports common industry standards, including the following:

ODBC	**ActiveX**	**SGML**
ODMA	Notes/FX 1.1	NotesFlow
MAPI	VIM	TAPI
HTML	SNA	AS400

Top

Microsoft Office 97 Applications Programmability Enhancements

In addition to the major changes listed earlier in this document, numerous other enhancements make developing with Microsoft Office 97 even easier. On the following pages are descriptions of those enhancements, organized by product.

Microsoft Access 97
Microsoft Excel 97
Microsoft Outlook 97
Microsoft PowerPoint 97
Microsoft Word 97

Top

Comments? Suggestions? Write us at MSDN@microsoft.com.
© 1998 Microsoft Corporation. All rights reserved. Legal Notices.
Last Updated: June 26, 1998

Best experienced with
Microsoft Internet Explorer
Click here to start.