
The Ultimate Source Code Library and Repository for
Microsoft Visual Basic 6, Access, Office, and VBA Developers

Total Visual
SourceBook™

For Microsoft® Office
and Visual Basic 6.0



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.

Redistributable Components

Only the Library Code included in the SOFTWARE may be redistributed. Library Code is defined as the programming code that is available when using the SOFTWARE program’s Code Explorer interface. You are specifically prohibited from redistributing any other file or part of the SOFTWARE.

Use and Redistribution Rights

FMS grants you the right to distribute the Library Code 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 the library code, in whole or in part, in any form that does not add primary and substantial value to the library code.
3. You do not distribute the library code in an application that is similar in nature or purpose to the SOFTWARE.
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 Visual SourceBook from FMS, Inc." and that you do not claim ownership of the SOFTWARE.

Publishing Library Code

You are prohibited from publishing library code on any medium, including, but not limited to, electronic mail, online services, internet newsgroups, and printed matter, except when library code is distributed in an application that adds substantial value to the library code and that application is not similar in nature or purpose to the SOFTWARE.

The Library Code may not be included in freeware or shareware applications in a non-compiled state.

Other Limitations

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, the recipient agrees to the terms in this SOFTWARE License, you remove all installations of the SOFTWARE, and you have not used any of the Library Code in an application. Ownership transfers must be reported to FMS, Inc. in writing.

Technical Support for Library Code

Because of the complexity of software development, FMS does not provide technical support related to the use of the Library code in your applications. Technical support is available for issues you find on the Code Explorer application and issues with the included Library Code and associated documentation.

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.

Total Visual SourceBook is copyright © by FMS, Inc. All Rights Reserved.

Microsoft, Access, Word, Excel, PowerPoint, Outlook, SharePoint, Visual Basic, Visual Basic for Applications, Visual Studio, Bing, and Windows are registered trademarks of Microsoft Corporation.

All other trademarks are trademarks of their respective owners.

Acknowledgments

We would like to thank everyone who contributed to make Total Visual SourceBook 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 Visual SourceBook, including:

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

Table of Contents

Chapter 1: Introduction	3
About Total Visual SourceBook	4
Product Highlights	4
Enhancements in This Version.....	6
Enhancements in Previous Versions.....	9
Visit Our Web Site	10
Chapter 2: Installation and Startup	13
System Requirements.....	14
Installation Overview.....	14
Upgrading from a Previous Version.....	14
Installing Total Visual SourceBook.....	15
Supporting Multiple Developers	17
Uninstalling Total Visual SourceBook	17
Chapter 3: Code Style, Conventions, and Usage	19
Coding Overview	20
Consistency.....	20
Dependencies	20
Avoiding Conflicts with Your Code	22
Naming Conventions	22
Error Handling	24
Structured Programming Style and Formatting	26
Other Issues.....	27
Example Code.....	28
Chapter 4: Code Explorer	31
Running the Code Explorer.....	32
Using the Code Explorer	34
Toolbar Icons	35
Code Explorer Windows	38
Getting Code into Your Project.....	43
Finding Code with the Search Tool.....	47
Using the Search Results	50
Creating and Using Bookmarks.....	50
Chapter 5: Administrative Tools and Options	53
Setting Options.....	54
Code Databases Tab	55
Error Handling Tab	56
Export Tab	60
Settings Tab	62
Syntax Highlighting Tab	63

Administrative Features	64
Chapter 6: Code Repository	67
Code Repository Overview	68
Importance of Backups.....	68
Multi-User Issues.....	69
System-Maintained Information	71
Adding New Items	72
Importing Code	80
Saving Classes, Modules, and Procedures.....	82
Editing Items.....	84
Deleting Code	84
Chapter 7: Source Code Exchange.....	85
About Source Code Exchange	86
Creating an SCE Package	86
Importing an SCE Package.....	87
Importing Data from another Code Library	89
Chapter 8: Product Support.....	93
Troubleshooting	94
Web Site Support	94
Technical Support Options	95
Contacting Technical Support	97
Index	99

Welcome to Total Visual SourceBook

Thank you for selecting Total Visual SourceBook, the most popular professional source code library for VBA and Visual Basic 6.0 developers. Total Visual SourceBook supports VB6/VBA developers regardless of platform, so whether you are in Access, VB6, Excel, Word, Visio, or any other VBA host, the program is available to make you more productive.

Total Visual SourceBook is developed by FMS, the world's leading developer of products for Microsoft Access and Visual Basic 6. In addition to Total Visual SourceBook, we offer a wide range of products for Microsoft Access and VB6 developers, administrators, and users:

- [Total Access Analyzer](#) (database documentation)
- [Total Access Admin](#) (database maintenance control)
- [Total Access Components](#) (ActiveX controls)
- [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 Zip Code Database](#) (city and state lookup lists)

Visit our web site www.fmsinc.com for more information. We also offer Sentinel Visualizer, an advanced data visualization program that identifies relationships among people, places and events through link charts, geospatial mapping, timelines, social network analysis, etc. Visit our Advanced Systems Group at www.fmsasg.com for details.

Please make sure you 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 Visual SourceBook.



Luke Chung
President

Chapter 1: Introduction

Total Visual SourceBook is the world's most complete collection of truly useful code for Microsoft Office and Visual Basic developers. In addition to providing a rich code library, Total Visual SourceBook acts as a code repository, making it easy to manage your code and share it among your team. This chapter introduces the product—use it as your starting point to find out more about the features and highlights, and see how the rest of the manual is organized.

Topics in this Chapter

- **About Total Visual SourceBook**
- **Product Highlights**
- **Enhancements in This Version**
- **Enhancements in Previous Versions**
- **Visit Our Web Site**

About Total Visual SourceBook

Total Visual SourceBook is the most complete collection of fully tested and consistently documented Visual Basic, Access, Office and VBA source code. The source code we've included comes from our dedicated staff of Visual Basic, Office, and VBA experts and represents many years of effort. You'll find consistent and original code that's not available anywhere else, and it's all available for you to inspect, modify, and incorporate into your projects royalty-free.

Best of all, our exclusive Code Explorer makes it easy to find the code you need, add your own code, securely share your code with other developers, and even package and ship code to others.

Product Highlights

Extensive Code

With 55 categories, 220+ classes and modules, 3400+ procedures, and 120,000+ lines of code, plus 21,000 lines of example code, every major area of application development is covered:

- Access Databases
- Access Startup Routines
- Access Forms and Reports
- Access Tables, Records and Fields
- Office Automation for Access, Excel, Outlook, PowerPoint, and Word
- ActiveX Controls
- Database: ADO, DAO and ODBC
- Birthdays and Anniversaries
- Business Day Math
- Clipboard Operations
- Compression Algorithms
- Data Type and Unit Conversions
- Date and Time
- Geospatial Addresses and Distance
- Geospatial Mapping
- HTTP and FTP
- Internet Search
- Math Routines
- Abstract Data Types
- Data Integrity
- Error Handling
- File and Disk Operations
- Forms and Controls
- Fonts
- Graphics
- Help and INI Files
- Multimedia Extensions
- Password Encryption
- Printing
- Registry
- Searching and Sorting
- Soundex and Metaphone
- SQL Server
- String Handling and Parsing
- System Information
- Visual Basic 6 Environment
- Windows Common Dialogs

- Statistics and Probability
- XML Files

And much more!

Ideal for All Levels of Expertise

Regardless of whether you are a seasoned veteran or new to Visual Basic, VBA, or Office development, you will find the source code available in Total Visual SourceBook to be extremely helpful. Novice and intermediate developers will find it easy to solve basic programming tasks and learn new techniques. Advanced and expert programmers benefit from the product's advanced classes to support high-level database operations, hard-core API functionality, Windows graphic functions, and more. And because the source code is open and available for modification, experts can customize the code to their specific needs.

Designed Specifically for VBA and Visual Basic 6.0

Total Visual SourceBook was designed from the ground up for use with Visual Basic 6.0 and the integrated VBA environments in Microsoft Office 2000 through 2013. It offers full integration with the Visual Basic Integrated Development Environment (VB IDE) and runs either from the Visual Basic Add-ins menu or as a standalone program.

You can select the code you want and insert it directly into your project. You can also export the code to the Windows Clipboard, a file, or open it with a text editor (such as Windows Notepad).

The program's Export dialog shows all of the existing forms, classes, and modules in your current project. To insert code, simply tell Total Visual SourceBook where you want the code to go. The Code Explorer also fully supports Visual Basic project groups.

Add Your Own Code to the Code Repository

Total Visual SourceBook allows you to add your own code to its repository and make it available to your entire development team. You can insert your code into existing topics and categories or create new ones. You can also add descriptions, comments, and examples to your code. The Code Repository even tracks the author name, creation date, and modification dates.

Standardize Error Handling

Total Visual SourceBook includes default error handling for every code item to prevent your applications from crashing with unhandled run-time errors.

The built-in error handling code identifies the name of the procedure where the error occurs, and displays the error code and message.

Of course your application's requirements or programming style may require error handling that's different from the default. Total Visual SourceBook allows you to update the error handler in the included code. See **Error Handling** on page 56 for details.

All Code Is Fully Documented

The source code in Total Visual SourceBook is extensively documented with reference material describing how to use the code. There are also overviews of various areas of programming technology in Visual Basic, complete with references to other sources of information (such as computer books and web sites). Additionally, the product uses a consistent style of documentation and commenting throughout.

Powerful Search Engine

Finding the code you need is easy—just browse through the categories in the Treeview to see what is available. If you need a more powerful search mechanism, use the Code Explorer's Search Engine to find code by keyword or phrase, or locate by author name, or creation and modification dates.

Enhancements in This Version

In 1996, we introduced Total Access SourceBook for Microsoft Access developers. The following year, we introduced Total VB SourceBook for Visual Basic 5 developers. Since then, the programming languages and developer communities have overlapped more and more. In 2000, we merged the two popular source code libraries and incorporated new code and new features. Since then, we have released several new versions to provide new code and program features.

The latest version of Total Visual SourceBook builds on our knowledge and experience and includes the most up-to-date and complete line of code for VBA and VB6 developers.

Total Visual SourceBook 2013, Version 15

Total Visual SourceBook 2013 introduces support for Microsoft Office 2013 and 2010 as well as Microsoft Office 2007, 2003, 2002 (XP), 2000, and Visual Basic 6. Additionally, it includes several enhancements from previous versions of the product, including:

Code Browser Enhancements

The browser is enhanced to improve the way code is managed:

- Category editing lets you move a category to another topic
- Module/class editing allows you to reassign it to another category, including categories in another topic
- New code and edit dialogs can be resized and maximized to simplify viewing and editing your code, notes, and examples
- When viewing code, the Details tab shows the editor name with date and time of creation and modification
- Toolbar and menus are simplified to edit the currently selected item

Select Default Error Handling Code

Under Options, error handling default settings can be selected for the error handling module or class you add to your application. This simplifies the specification of the Error Enabler, Re-Enabler, and Handler, plus the use of a global constant to turn error handling off or on.

User Interface Enhancements

- More modern user interface with modern looking controls, improved graphics and Windows themes
- Most screens are enlarged, display more data with more readable fonts, use larger fonts sizes, and show longer dropdown lists
- Improved support for shortcut keys on forms and menus
- New manual and help file

Import/Export and Source Code Exchange

- Larger Export and Import form to display the Module list with wider column widths
- Import user code from Total Visual SourceBook 2007 and assign them automatically to the same topics and categories
- Source Code Exchange feature revamped to support non-standard and Unicode characters
- Source Code Exchange Export form retains the previous entries

All Code Includes 32 and 64-bit Support

A 64-bit version of Microsoft Office was introduced with Office 2010. VBA code that uses Windows API calls, DLLs, or ActiveX controls, need to support the 64-bit version when running in the 64-bit version of Office. Code that referenced the 32-bit version will not compile in the 64-bit version. Similarly, 64-bit code won't run with the 32-bit versions.

For all the relevant modules in Total Visual SourceBook 2013, 32 and 64-bit support is included through compile time #IF statements. This allows the same code to be run in both versions so you can create solutions without worrying about which version your users are running today or tomorrow.

Code Enhancements

New classes and modules were added and existing code updated to add new functionality and support Access and Office 2013 and 2010.

Here is some of the most significant new code:

- Access database startup routines
- Access data exports to Excel
- Access table modifications
- Access attachment, hyperlink, and multivalued fields
- Enhanced error handling with a new module specifically for Microsoft Access with advanced error logging and email options
- Age, birthday and anniversary calculations
- Email address format validation
- Excel data import, cell and range formatting, pivot tables
- Converting location addresses to latitude and longitude
- Google Earth mapping routines
- Google Maps and Bing Maps locations and directions
- Geographic distance between two points
- Hex and character translators with URL support
- Internet search for Bing, Google, YouTube, and Yahoo
- MapQuest mapping
- Outlook address and contact lists
- Phone number letter conversions
- PowerPoint theme, data import and slide formatting
- Shorten Internet URLs with Bit.ly and TinyURL converters
- Stock market quotes
- Tables linked to SharePoint lists
- Twitter messages
- Windows File Dialog masks and using API vs. VBA approaches
- Word merge
- XML file creation and reading, plus Unicode support
- Zip file creation and extraction (compressed files)

User Code Can be Stored in a Microsoft SQL Server Database

Total Visual SourceBook has always allowed you to save your own code and share it with others by storing it in a Microsoft Access database. You can now upsize that database to SQL Server and share your code from there. You can use your own SQL Server host or a cloud provider like SQL Azure. If you and your fellow developers can connect to your SQL Server database, you can easily search, use, and update your shared code.

Improved Setup Program

The Setup program is improved to support user or machine installation options respecting user access control (UAC) permissions, and online activation of the product.

Enhancements in Previous Versions

Total Visual SourceBook 2007, Version 12

Total Visual SourceBook 2007 introduces support for Microsoft Office 2007 as well as Microsoft Office 2003, 2002 (XP), 2000, and Visual Basic 6. Additionally, it includes several enhancements from previous versions of the product, including:

Browser Enhancements

The browser has been enhanced with the following features:

- Streamlined user interface and Vista support
- New documentation, including a new CHM help file
- Enhanced Error Handler options that make it easy to implement your own error handler in FMS code

Code Enhancements

New code has been added for a variety of topics, including new code to support Access and Office 2007. The new topics covered include:

- Access 2007 Jet features such as multi-value and attachment fields
- Access/Office 2007 Ribbons
- Enhanced support for ADODB and ADPs
- Jet database management, including security and encryption
- Copying tables and records
- Sending and retrieving Twitter messages
- Working with Visual SourceSafe
- Working with Windows XP and Vista folders

- Access Application startup routines
- Zip code lookup for city and state names
- Access spell checking
- Working with Outlook Calendar appointments
- Working with Outlook Notes
- Improved business day and holiday math
- Support for HTML and CHM help files
- Improved mouse support
- Improved printer support
- Windows Media Player

Existing code has been significantly enhanced to use modern coding techniques, and to include more complete documentation and examples.

Backward Compatible

The code you added to the code repository of Total Access SourceBook, Total VB SourceBook, and previous versions of Total Visual SourceBook can be imported into this version of the program.

Visit Our Web Site

FMS is constantly developing new and better developer solutions. Total Visual SourceBook 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 Visual SourceBook 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

- Visit our blog at: <http://blog.fmsinc.com>
- Follow us on Twitter: <http://www.twitter.com/fmsinc>
- Like our [Facebook page](#)
- Visit our technical support forums at: <http://support.fmsinc.com>

Chapter 2: Installation and Startup

Total Visual SourceBook comes with an automated setup program to get you up and running as quickly as possible. This chapter describes the system requirements, installation steps, configuring it for multiple developers to share their code, and instructions for uninstalling.

Topics in this Chapter

- **System Requirements**
- **Installation Overview**
- **Upgrading from a Previous Version**
- **Installing Total Visual SourceBook**
- **Supporting Multiple Developers**
- **Uninstalling Total Visual SourceBook**

System Requirements

Total Visual SourceBook requires:

- Windows XP or later including Vista, 7 or 8 (32 or 64-bit version)
- Sufficient hardware to run Microsoft Office, Access and/or Visual Basic 6.
- 30 MB free disk space to install the product, and additional disk space to store your code databases.

Installation Overview

While the Setup program is designed to make installation as simple as possible, you should take a few moments to read through this chapter in order to determine the configuration that best suits your needs.

Administrator Rights Required for Installation

Total Visual SourceBook is installed through a professional setup program that conforms to Windows install and uninstall standards. Total Visual SourceBook requires the installation of ActiveX controls and DLLs, so Administrator rights are required. You can install the program so that it works for the current user or everyone on the machine.

Multiple Developer Support

Total Visual SourceBook offers a great deal of flexibility in how it stores code and makes code available to multiple developers. By default, the database storing your code is located in the same folder where Total Visual SourceBook is installed. To share code across multiple developers, this database should be on a network folder accessible to all the developers on your team. After you install the program, you can point to the database under the program's Tools, Options menu.

Upgrading from a Previous Version

Upgrading from Total Visual SourceBook

If you have a previous version of Total Visual SourceBook, upgrading to the new version is simple:

1. Install the new version to a new folder (do not overwrite your existing version).
2. Run Total Visual SourceBook 2013
3. Import the user code from the prior version by going under the menu: File, Import, Another Code Library, and choose Total Visual SourceBook 2000-2007

Your code is now part of your Total Visual SourceBook 2013 application.

Alternatively, if you want to share the user code with the legacy version of Total Visual SourceBook, you can point to the existing user database. Under the menu Tools, Options, Code Databases tab, select your existing user code. See page 55 for more details.

Upgrading from Total Access SourceBook/Total VB SourceBook

If use Total Access SourceBook (for Access 2000 or earlier) or Total VB SourceBook (for Visual Basic 5 or earlier), you can import your user code into Total Visual SourceBook.



Tip

You should not uninstall Total Access/Total VB SourceBook until you verify that all user code has been successfully imported. Once your old code is in Total Visual SourceBook, you can uninstall the old version, and delete any databases or other files that may remain.

For complete information about importing code from previous versions of SourceBook, or other code libraries, see **Importing Data from another Code Library** on page 89.

Installing Total Visual SourceBook

Total Visual SourceBook is installed using a professional setup program that conforms to Windows standards.

The Total Visual SourceBook Code Explorer consists of a executable program, various DLL and OCX files, and entries in the system registry. This program must be installed on your local computer to run it. You cannot install it on a network drive and have other users to run it directly from there. Because of its dependencies, administrator rights are needed to install the program.

1. Run the setup program with Administrator rights.
2. When prompted, enter your registration information and product serial number.

3. Specify whether you want to install it just for you (the current user) or anyone who uses the PC (machine install). The latter allows an administrator to install the program so all non-administrator users on the machine can run it.
4. Specify the destination folder for the files. This folder should not be readonly since the default location for the user database is here.
5. After the installation is completed, read the readme file for any information that is not included in the manual or help file.

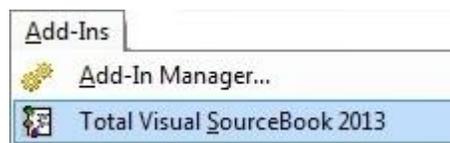
Running Total Visual SourceBook

Total Visual SourceBook can run from several locations. It can run as a standalone program from the Windows menu:



Total Visual SourceBook Shortcut from the Windows Programs

Total Visual SourceBook is also available as an add-in within your Visual Basic 6 and VBA/Office code editor called the Integrated Development Environment (IDE):



Total Visual SourceBook from the IDE Add-Ins Menu

Loading Total Visual SourceBook as a VBA Add-in for Each User

The VBA IDE does not support the loading of add-ins for the entire machine. Each user needs to load it separately. If you install Total Visual SourceBook for the machine, each user other than the one who installed it, needs to run a file to add it to their VBA add-in menu.

A registry file (VBA-Addin.reg) is in the folder where Total Visual SourceBook is installed. Each user simply needs to run that file to add Total Visual SourceBook to their add-in menu. The file updates the user section of the registry, so administrator rights are not needed.

Note that this is not necessary for VB6 which supports machine installation of add-ins.

Supporting Multiple Developers

Total Visual SourceBook contains two databases to store the source code it displays:

- The built-in, internal code (the source code that ships with the product) is stored in the FMS Code database: TVSBIC.MDB.
- The code you add is stored in the User Code database: TVSBUC.MDB.

One of the most powerful features of Total Visual SourceBook is that you can create and use multiple Code Databases. By default, the User Code database is stored in the same folder Total Visual SourceBook is installed. You can rename it or move it to any location.

The Code Database files are specified under Tools, Options on the **Code Databases Tab** on page 55. Everyone on your team can point to the same file on a shared network drive to easily share their valuable work.



Note that each user who wants to access a shared Code Database must have a separate licensed copy of Total Visual SourceBook. Volume licenses are available—contact FMS for more information.

Shared, Consistent Error Handling

When sharing your team's source code, you should also specify the error handling so all the code exported by Total Visual SourceBook uses your team's standards. This is also set under Tools, Options on the **Error Handling Tab** on page 56.

Uninstalling Total Visual SourceBook

Preserve Your Source Code

Before removing Total Visual SourceBook from your system, take steps to ensure any code that you need remains available to you after the program is uninstalled. By default, it's named TVSBUC.MDB in the folder where Total Visual SourceBook is installed.

The Total Visual SourceBook uninstall process does not delete your User Code database. This allows you to access your code after uninstalling simply

by reinstalling the program. However, if you plan to uninstall Total Visual SourceBook and not reinstall it in the future, you will not be able to retrieve your code from the code databases. In this case, be sure to export your code to an external format (such as a VB/VBE project or text files) so that you can access your code without having the Total Visual SourceBook program installed.

Uninstalling

Total Visual SourceBook conforms to Windows installation and removal standards. To uninstall Total Visual SourceBook:

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

The uninstall program DOES NOT remove the FMS or User Code databases. The User Code database is your code repository and contains code you added. The FMS Code database is not deleted since multiple users could be pointing to it. To completely remove all of Total Visual SourceBook's files, manually delete the code databases.

Chapter 3: Code Style, Conventions, and Usage

All developers have their own style of writing code. In creating Total Visual SourceBook, we tried to take this into account. The style, comments, and error handling for each procedure are what we consider to be the best mixture of common practices, while allowing you to use and customize the code with the least disruption to your programming system.

Topics in this Chapter

- **Coding Overview**
- **Consistency**
- **Dependencies**
- **Avoiding Conflicts with Your Code**
- **Naming Conventions**
- **Error Handling**
- **Structured Programming Style and Formatting**
- **Other Issues**
- **Example Code**

Coding Overview

All experienced programmers have their own ideas about good programming style. Although certain standards exist within programming communities, organizations and individuals often develop their own habits and practices based on company standards, consistency with existing code, or even the personal preferences of influential team members.

Accordingly, it is impossible for the code style in Total Visual SourceBook to satisfy 100% of developers. However, the guidelines we use are consistent with a majority of development shops today, with respect to naming conventions, structured programming techniques, and object-orientation.

This section discusses the programming style and standards used in Total Visual SourceBook. It gives you a better idea of the reasoning behind the choices we made, and prepares you to make modifications to suit your programming style.

Consistency

We placed a high premium on consistency in the style and techniques of the code in Total Visual SourceBook. After you use one procedure or class from Total Visual SourceBook, it should be easy to find and use others. You will find that the naming conventions, commenting, and functional behavior is consistent among all of the code in Total Visual SourceBook.

Our assumptions and standards are described in this section, so you can easily convert from our standards to yours if desired.

Dependencies

One of the primary goals of Total Visual SourceBook is that the source code provided should have no functional dependencies on any run-time components that we ship, such as OCXs, helper DLLs, or OLE type libraries. We are delivering source code, not binary components. The source code is inserted directly into your project. You do not need to think about distributing any other objects with your application.

In fact, one of the main benefits of the code in Total Visual SourceBook is that it allows you to eliminate some dependencies. For example, the code includes routines that replace the functions of the COMDLG32.OCX control, or the multimedia player OCX.

Code Dependencies

Every module and class is independent and can be inserted into your project and immediately work and compile. You do not need to use special compiler directives, create references, or try to track down dependencies when you insert a module or class into your project. The module or class acts as a true black box.

For modules, you may want to only grab a single procedure rather than the whole module. In most cases, this works, but there are cases where dependencies exist.

Declarations Dependency

The most common procedure dependency is when it uses a Windows API call, type, enum, or constant defined in the module Declarations section. When a procedure has a dependency on the Declarations, Total Visual SourceBook keeps track of it and lets you automatically export it along with the procedure.

Procedure Dependency

In some cases, procedures may call other procedures within the same module. The Code Explorer handles this by displaying all code dependencies on the Details tab (as described on page 42), and by allowing you to export the dependencies along with the procedure.

Class Dependencies

All of the class modules in Total Visual SourceBook are fully self-contained. We do not have any classes that call other classes or external procedures in Total Visual SourceBook.

In some cases, Total Visual SourceBook procedures in modules are also duplicated in classes as private helper functions. This results in slight duplication of code, but it helps us achieve the no-dependency rule for classes.

Other Dependencies

Naturally there are some dependencies on features that are external to Total Visual SourceBook itself. For example, a certain procedure might require that you create a reference to the Microsoft ADO library, or that you have a certain version of the Windows common controls installed on your machine. For example, certain functions may work on Windows 9x, but not on Windows NT. Consult the Dependencies information on the Details tab and the code Notes for complete details.

Avoiding Conflicts with Your Code

Because we limit dependencies in our code, it is unlikely that adding Total Visual SourceBook procedures or classes to your project will create conflicts with your existing code. All Windows API declarations and constants are declared as “Private,” so they are only scoped to the class or module where they occur. If a procedure takes an argument that is a constant, the constant value is provided via a uniquely named enumerated type rather than via a public constant.

The only likely conflict is with the name of the procedure or class itself. If your code contains an existing procedure with the same name as a procedure that you import from Total Visual SourceBook, your code will not compile. You can resolve name conflicts by saving the Total Visual SourceBook procedure under another name in the browser, and then exporting the new procedure.

Naming Conventions

Opinions on the proper use of naming conventions vary widely among developers. Most developers adhere to one of many industry recognized naming conventions, while others use their own personal or organizational conventions. Still others strongly believe that there should be no naming conventions at all. As with many other areas of programming, the true value of using naming conventions is the consistency they provide, rather than the merits of any one particular standard.

The style used in Total Visual SourceBook is a variation on the recommendations that Microsoft makes in the “Visual Basic Coding Conventions” chapter of the VB Programmer’s Guide. These include naming standards for controls, objects, menus, properties, types, constants, and other standard objects. The naming conventions also indicate scope, and whether or not the variable is part of an array or a collection.

Variable Type	Tag	Example
Boolean	f	flsLoaded
Byte	byt	bytOffset
Currency	cur	curAmountDue
Date	dat	datExpires
Double	dbl	dblOffset

Variable Type	Tag	Example
Enumerated Type definition	Enum	EnumJetObjectTypes
Enumerated Type use	e	eObjectType
Integer	int	intCounter
Long Integer	lng	lngFrequency
Module	mod	modApplication
Object	obj	objExcel
String	str	strLastName
Variant	var	varObject
Array	a	astrNames
Constant	c	cintMaxValue

For controls and other objects available in Visual Basic, the following naming conventions are used:

Object Type	Tag	Example
Class	C	CEncryption
Control	ctl	ctlListBox
Form	frm	frmMainMenu

For database objects defined in the DAO, RDO, or ADO interfaces, the following naming conventions are used:

Object Type	Tag	Example
Command (ADO)	cm	cmUpdate
Connection (ADO)	cn	cnnPubs
Container	con	conForms
Database	db	dbCurrent
Document	doc	docTmp
Field	fld	fldLastName
Parameter (ADO & DAO)	prm	prmCustID
QueryDef	qdf	qdfSelect
Recordset (ADO & DAO)	rst	rstOrders
TableDef	tdf	tdfCustomers

To indicate the scope of a variable, the following scope tags are prefixed to the variable name:

Scope	Tag	Example
Module level variable	m	mstrLastName
Public variable	g	gstrLastName
Local variable to support property procedure values	m_	m_datCurrentDate

Windows API Declarations

Wherever the code contains a declaration to a procedure in the Windows API, the variable names and naming conventions published by Microsoft in the API Programmers Reference are used instead of the above naming conventions. This ensures that our declarations match the Windows API.

Error Handling

All Total Visual SourceBook code includes default error handling code to protect against unhandled application crashes. Additionally, this error handling code is fully customizable—you can update it to match your style or requirements. See **Error Handling Tab** on page 56 for details.

Default Error Handling

When using code from Total Visual SourceBook, we don't want you to have to deal with complex error handling setup and maintenance. However, we strongly feel that every non-trivial subroutine, function, and property should include error handling. Therefore, all Total Visual SourceBook code includes what we consider the minimal acceptable error handling. The default error handling code is only intended to stop your application from crashing, not to act as an exit point during normal operation. The error handler accomplishes the following:

- Identifies the name of the procedure with the error
- Displays the error code and error message
- Prevents your program from crashing with a fatal run-time error

The following is the structure of the default error handler:

```

Function MyName(strParam As String) as Boolean
  ' Comments: Comments about MyName
  ' Params : strParam      Parameter definition
  ' Returns : True or False
  ' Source : Total Visual SourceBook

  On Error GoTo PROC_ERR

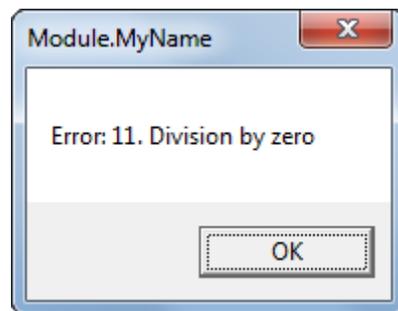
  Debug.print 1 / 0

PROC_EXIT:
  Exit Function

PROC_ERR:
  MsgBox "Error: " & Err.Number & ". " & Err.Description, , "Module.MyName"
  Resume PROC_EXIT
End Function

```

When this function is called, it generates a run-time error:



Generic Error Handler Results

The error handler is established at the beginning of the procedure by using “On Error GoTo PROC_ERR.” The procedure error label is always called PROC_ERR, so it is easy to locate in the code.

In fact, labels and the “GoTo” statement in Total Visual SourceBook procedures are only used for error handling. Rarely, if an error condition is detected that is not an actual run-time error, the code may explicitly branch to the “PROC_EXIT” label. See the section below on structured programming style for more information on this exception.



The same style of error handling is used in standalone procedures and functions, as well as in procedure and property members of class modules. This is done for consistency and to provide a minimum fallback level of error handling for every item.

However, you may prefer to have errors in classes raised back to the caller, rather than having the class members process the error handling with message boxes. This is especially important for classes that are exposed from automation servers where there is no “user” available to respond to the message box. To have the class caller handle run-time errors, strip out the error handling in the class members altogether, or replace it with custom error handling appropriate to your application.

Customizing Error Handling

There are as many approaches for error handling as there are many programmers. The error handling included in Total Visual SourceBook code provides the minimum acceptable error handling, but you may prefer to use error handling that matches your own programming style. Total Visual SourceBook makes it simple to replace the default error handling code with your own. See **Error Handling Tab** on page 56 for details.

Structured Programming Style and Formatting

Total Visual SourceBook’s procedures, functions, and class methods follow these generally recognized standards for structured programming style:

- Each subroutine or function has only a single exit point. “Exit Sub” and “Exit Function” are not used, except in the context of a single defined exit label.
- The “GoTo” statement is not used except to enable a module-level error handler. Similarly, the “GoSub” statement is not used. In rare cases, an unconditional branch to the procedure exit label may be used in order to avoid deeply nested conditional code.
- Line labels are not used except for procedure handling exit points.
- There is no reliance on public or global variables for procedures. A class may have static shared data associated with it, but procedures and functions have no external data dependencies.

We use standardized formatting for items such as indenting, where local variables are declared, how in-code comments are formatted, and how error handling is enabled. While the standards may not match yours, our consistency should make it easy for you to understand and use the code.

Other Issues

By Reference versus By Value

When information must be returned from a procedure, it is done with the return value of a function rather than by modifying a passed argument by reference. Exceptions to this rule include cases where the procedure populates a passed array or type structure, or when more than one item of information must be returned.

Class versus Module

In many cases it is a judgment call whether to use a standard module to group related functions or to create a class module. These are the guidelines we use to determine that a class should be used rather than a standard module:

- More than one procedure requires access to common data (for example, a pointer to an object variable such as a form, control, or database object)
- The procedure must remember “state” information between calls
- You may want to have more than one instance of the code be running at the same time without interfering with other instances

Commenting

In addition to the reference material available in the Total Visual SourceBook Code Explorer, all of the procedures include extensive in-code comments.

Each Declaration, subroutine, function, or class property procedure includes a structured comment block indicating the purpose of the function, a description of its arguments, and return values.

Procedures also include in-line comments to clarify the purpose of tricky or non-obvious code.

Administrative Comments

In addition to the comments that explain the purpose and workings of the code, every procedure contains a line indicating that the source code comes from Total Visual SourceBook:

```
' Source      : Total Visual SourceBook
```

This line is included so you always know which code in your project came from Total Visual SourceBook. While you do not have to keep this comment line in your project, if you take it out, you may have difficulty determining which code came from you or Total Visual SourceBook.

Technical Level

Some of the routines in Total Visual SourceBook make use of Windows API calls to provide functionality that is not available natively in VB6/VBA itself, or to replace standard VB6/VBA functionality with a more efficient API method.

In most cases the code is self-explanatory, but we have not attempted to provide alternate documentation for the Windows API. If you have questions about the techniques used, please refer to Microsoft's documentation.

Although our code uses some fairly advanced techniques, nothing in our code is wildly "hard core" or over the edge in terms of maintainability, reliability, and level of difficulty. We do not hack OLE type libraries, or use any other bleeding edge techniques. There are certainly routines that could cause problems if used incorrectly, but that is true any time you use Windows API calls or calls into other external DLLs.

Example Code

The code solutions in Total Visual SourceBook include example code to show how the code is called, and the syntax used. All example code is accessible directly in the Code Explorer by using the Example Tab (see page 42 for more information). An example exists for every class and module in Total Visual SourceBook.

To get you up and running as quickly as possible, we've implemented each example as a ready to run form.

To use the example code, follow these steps:

1. Create a new project in VB6 or the VBA host language you are using. While this step is not required, you may want to do this so you do not create example objects in one of your "real" projects.
2. Export the class or module you want to test into your new project by using the Total Visual SourceBook Export Dialog (explained on page 44).

3. Create a new form, document, or spreadsheet in your project. Don't worry about naming the object, since it is only a sample object in a sample project.
4. Follow any additional directions contained in the comment text at the top of the code. These directions usually require you to add one or more controls to your new form or document.
5. Paste the entire contents of the Example code into the new object's module and run it.

The following code shows a representative Example:

```
' Example code for modToolbar
'
' To try this example, do the following:
' 1. Create a new form
' 2. Create a toolbar named 'ToolBar1'
' 3. Create a command button named 'cmdTest'
' 4. Paste all the code from this example to the new form's module.
' 4. Run the form

Private Sub cmdTest_Click()
    Static fFlat As Boolean

    fFlat = Not fFlat

    ' Example of SetToolBarFlat and 'flat' mode
    SetToolBarFlat ToolBar1, fFlat
End Sub

Private Sub Form_Load()
    Dim btn As ComctlLib.Button

    With ToolBar1
        Set btn = .Buttons.Add(, , "Button 1")
        Set btn = .Buttons.Add(, , "Button 2")
        Set btn = .Buttons.Add(, , "Button 3")
        Set btn = .Buttons.Add(, , "Button 4")
    End With
End Sub
```


Chapter 4: Code Explorer

The Code Explorer is the user interface for Total Visual SourceBook and organizes the source code. Use the Code Explorer to find the code you need, export it to your application, and manage your own code.

Topics in this Chapter

- **Running the Code Explorer**
- **Using the Code Explorer**
- **Toolbar Icons**
- **Code Explorer Windows**
- **Getting Code into Your Project**
- **Finding Code with the Search Tool**
- **Using the Search Results**
- **Creating and Using Bookmarks**

Running the Code Explorer

Starting the Program

Total Visual SourceBook supports two modes of operation: Standalone and Add-In.

Standalone

In Standalone mode, you start the program directly from Windows. Use this mode when you do not have Visual Basic or your VBA host open and want to use Total Visual SourceBook on its own. From the Windows Start menu, select Programs, FMS, Total Visual SourceBook, Total Visual SourceBook.

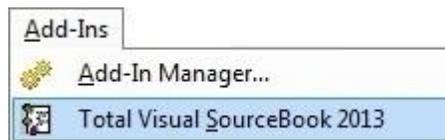
Alternatively, you can execute the file named TVSB.EXE located in the program folder you specified during installation.

Add-In

In Add-In mode, you start Total Visual SourceBook from within the Visual Basic integrated development environment. This is probably the most common way to start the program because, as an add-in, Total Visual SourceBook can export code directly into your VBA Project.

To launch Total Visual SourceBook as an add-in, open the VBA integrated development environment (IDE) for your application. For example, in Microsoft Access, open a new or existing module. In Microsoft Word, select Tools, Macro, Visual Basic Editor. Consult your product documentation on using the VBA environment in your particular application.

Select Add-Ins, Total Visual SourceBook from the VBA IDE Menu.



VBA IDE Add-Ins menu

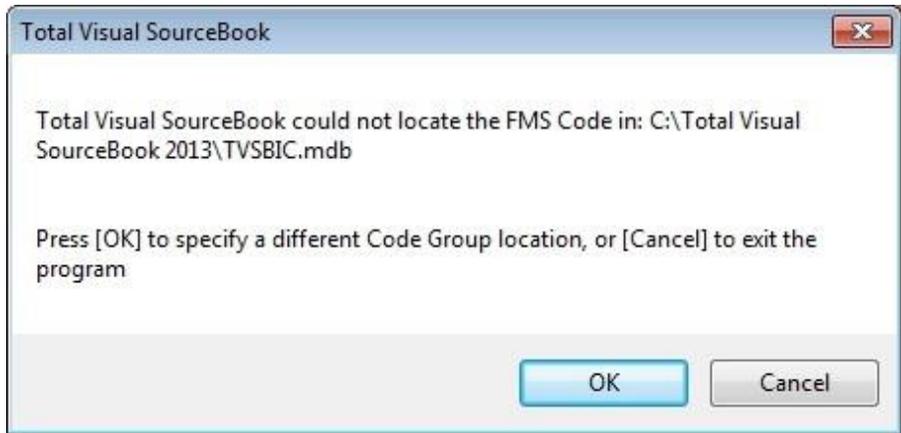
Please note that if you already have a standalone instance of Total Visual SourceBook open, this instance is used when Total Visual SourceBook is invoked from the add-in menu.

When using multiple instances of your host VBA application, each instance invokes a new copy of Total Visual SourceBook. This allows Total Visual SourceBook to work correctly with multiple instances of the application.

Using the Locate Code Group Dialog

When you start Total Visual SourceBook, the Code location you last used determines the code you see. The Code Group consists of the FMS Code and the User Code database containing the code you or your team members added.

If the last Code Group you used is not available when Total Visual SourceBook starts, the program displays the following message:



Locate Code Group Dialog

Click [OK] to open the Locate Code Group dialog and specify the directories for the FMS and User Code databases. This situation can occur for a number of reasons:

- The shared drive holding the Code Group is not currently available. For example, you are using a laptop computer that is not currently connected to the network. Use the Locate Code Group dialog to point to the folder where you installed Total Visual SourceBook on your machine. This allows you to continue using the program even if you are disconnected from the network drive. Of course, this means you won't have access to the User Code available in the shared Code Group.
- The Code Group was deleted or moved. Determine the location of the Code Group and select it with the dialog.
- The Code Group is corrupt or damaged. In such cases, restore a backup copy of the Code Group.

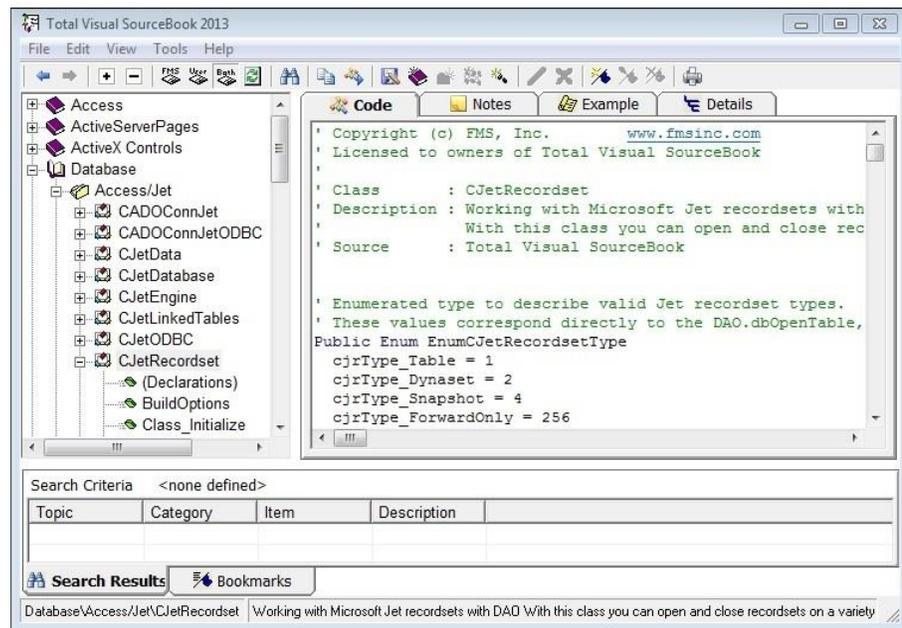
- Another user is running an administrative operation on the shared Code Group. This exclusively locks the databases. For example, while the “Compact Databases” (see page 65) operation is running, the Code Group databases are locked and access is disallowed. If this is the case, press the [Cancel] button on the Locate Code Group dialog and wait until the databases are available again.

Joining Other Code Groups

You can change the Code Group that the program uses by selecting Tools, Options from the menu and pressing the Code Group tab. For more information on this option, see page 55. For more information on Code Groups and how to use them effectively, see page 15.

Using the Code Explorer

When you open Total Visual SourceBook, the Code Explorer appears:



Code Explorer

This is where you do most of your work. Use the Code Explorer to find Total Visual SourceBook’s own procedures as well as to create, copy, and edit your own procedures. The Code Explorer interface is divided into three main areas:

- The **Treeview Window** shows all topics, categories, classes, modules, and individual procedures.

Forward

Also available from the View, Forward menu, this icon is available if Back was used. This feature moves to the next item in the history list.

Expand All

Also available from the View, Expand All menu, this feature expands the Treeview to display all items in all categories.

Collapse All

Also available from the View, Collapse All menu, this feature collapses the Treeview items so only Topics are visible.

Show FMS Code

Also available from the View, FMS Code menu, this setting causes the Treeview to show only the code that ships with Total Visual SourceBook.

Show User Code

Also available from the View, User Code menu, this setting causes the Treeview to show only code that you or your team members added to the Code Repository.

Show Both FMS and User Code

Also available from the View, Both menu, this setting causes the Treeview to show all the FMS and user code.

Refresh

Also available from the View, Refresh menu, this feature refreshes the Treeview by re-reading the contents of both the FMS code and User code databases. This is useful in multi-user installations when other developers change the contents of the database. If they add, edit, or delete items, Total Visual SourceBook does not automatically update the display to show the most current changes. The Refresh option ensures you are viewing the most current data.

Search and Export Icons

The second set of icons helps you find and extract the code you want.

Search

Also available from the Tools, Search menu, this icon opens the Total Visual SourceBook Search form to begin a search. For more information, see page 47.

Copy

Also available from the Edit, Copy menu, this feature copies the text displayed in the Information Window to the Windows clipboard. For more information, see page 44.

Export

Also available from the File, Export menu, this feature exports the item currently displayed in the Treeview to the clipboard, notepad, a text file, or Visual Basic. For more information, see **Exporting Code** on page 44.

Code Repository Icons

The third set of icons is for the Code Repository and helps you add and manage user code and categories. For more information, see page 68.

Save As

Also available from the File, Save As menu, this feature saves the item currently displayed in the Code Explorer Treeview under a new name. You can save either FMS procedures or your own procedures with this feature.

New Topic

Also available from the File, New Topic menu, this feature adds a new user Topic to Total Visual SourceBook.

New Category

Also available from the File, New Category menu, this feature adds a new user Category to Total Visual SourceBook.

New Class or Module

Also available from the File, New Class/Module menu, this feature adds a new user class or module to the Total Visual SourceBook code base.

New Member or Procedure

Also available from the File, New Method/Procedure, this feature adds a new class member or module procedure.

Edit

Also available from the Edit menu, this feature opens the Total Visual SourceBook Edit form to edit the currently selected item in the Treeview. It could be a topic, category, class/module, or procedure. You may only edit user-defined topics, not FMS items.

Delete

Also available from the File, Delete menu, this feature deletes a user item. Note that you cannot delete FMS items.

Bookmark Icons

The next set of icons helps you create and manage bookmarks. For more information on bookmarks, see **Creating and Using Bookmarks** on page 50.

New Bookmark

Also available from the Edit, New Bookmark menu, this icon creates a new bookmark for the selected item in the Treeview.

Edit Bookmark

Also available from the Edit, Edit Bookmark menu, this icon edits the current bookmark in the bookmark pane.

Delete Bookmark

Also available from the Edit, Delete Bookmark menu, this icon deletes the bookmark currently selected in the bookmark pane.

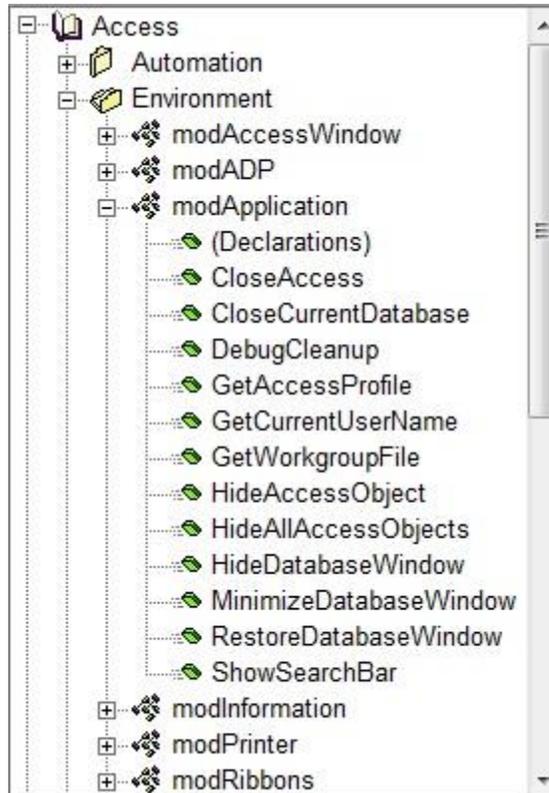
Code Explorer Windows

The Code Explorer is composed of three main windows:

- **Treeview Window**
Displays the code topics, categories, modules, and procedures in a hierarchical tree.
- **Information Window**
Shows the actual code, notes, and other relevant information on the selected code object.
- **Tracking Window**
Lists the Search Tool results and any Bookmarks you set.

Treeview Window

The Total Visual SourceBook Code Explorer Treeview is a hierarchical display of both FMS and user-written code. The Treeview makes it easy to find the code you want. All the code is organized into topics and categories.



Code Explorer Treeview

Topics

Each topic covers a major product or area of functionality, such as “Database” or “Access.” In this case, code pertaining to database use that is not specific to the product Microsoft Access is kept in the “Database” topic. Code that is specific to Microsoft Access (e.g. code dealing with Access Reports) is under the “Access” topic.

Categories

Within a topic, a Category organizes areas of functionality. For example, there is a “Forms” category under both the “Access” and the “Visual Basic” topic. Code in the “Forms” category under the “Visual Basic” topic has code that is specifically designed for use with Visual Basic forms. Code in the “Forms” category under the “Access” topic is designed to only work with Access forms.

Classes and Modules

Each category contains classes and/or modules. Each class contains its members (properties, methods, and procedures), and each module contains procedures. By clicking on each level, the next level expands, similar to Windows Explorer.

The following items are displayed in the Treeview:

-  **Topic** – A topic contains categories.
-  **Category** – A category contains classes and modules.
-  **Class** – A class implements properties and methods.
-  **Property** – A property is a property of a class.
-  **Member** – A member is a member of a class (sub or function).
-  **Module** – A module contains procedures.
-  **Snippet Module** – A snippet module is just the same as a regular module, except that it can contain any code snippets, text, notes, or information.
-  **Procedure** – A procedure is contained in a module.

To display a list of classes and modules in a selected category, double-click on the category in the Code Explorer Treeview. To show the list of procedures and properties available in a module or class, double-click on the item or press the [+] in the Treeview. To hide the list within an item, double-click again. You may also navigate through the list by using the arrow keys, and you may open or close the item by pressing the [Enter] key.

Total Visual SourceBook displays FMS code, user-written code, or both, depending on the setting of the View you have selected with the menu or toolbar.

The icons for user-written code are similar to those shown above, except that they are marked with a small blue “u” symbol in the lower-left-hand corner of the icon:

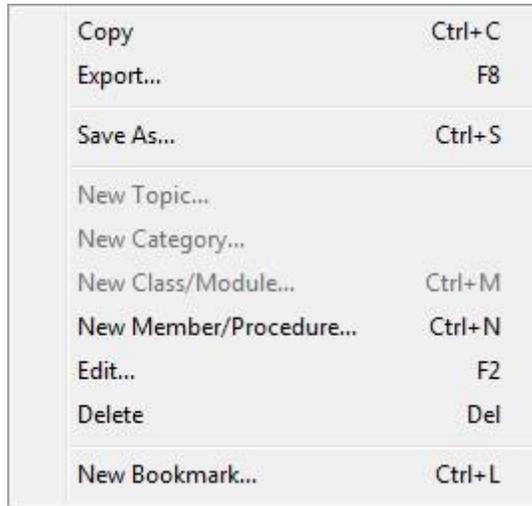


Example of User Code in the Treeview

When you select a particular item in the Code Explorer Treeview, the code is displayed in the Code Window to the right. If you select a category rather than a code item in the Treeview, a short description of the category is displayed in the Code Window.

Treeview Context Menu

As a convenience, a context-sensitive popup menu is available when you right click on the Code Explorer Treeview. This menu duplicates the relevant choices on the toolbar and menu. Menu selections not applicable to the currently selected item are disabled. See the toolbar section (page 35) for more information on each menu item.



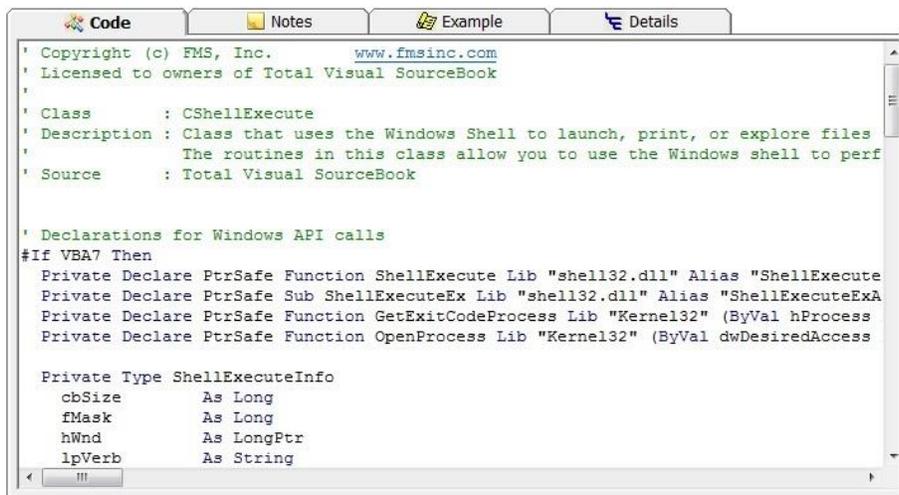
Treeview Right-Click Popup Menu

Information Window

The Information Window is a tabbed display containing details about the currently selected item. As you click on items in the Treeview, the Information Window updates to reflect the item you select.

Code Tab

The Code Tab displays the source code of the currently selected item:



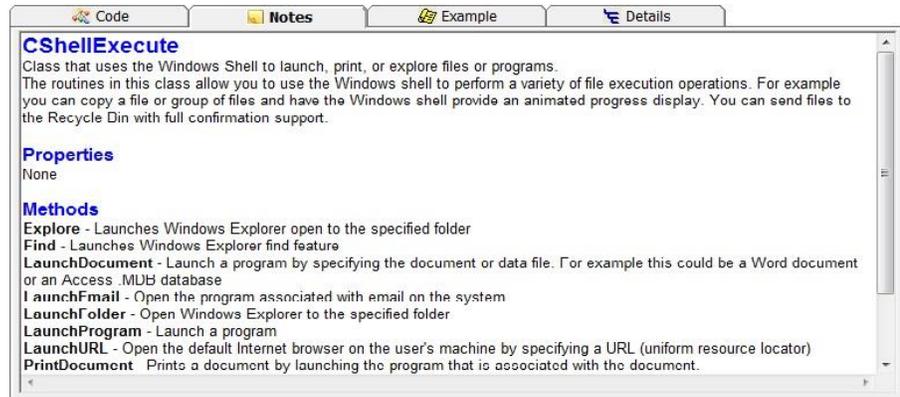
Code Tab

If the Code Explorer is currently on a topic or category rather than on a code item, the Code Tab is empty since code is not stored at the Topic or Category level.

Although you cannot edit code displayed in this window, you can select and copy it to the Windows clipboard. This is a convenient and quick way to copy a selected procedure without going through the File, Export process.

Notes Tab

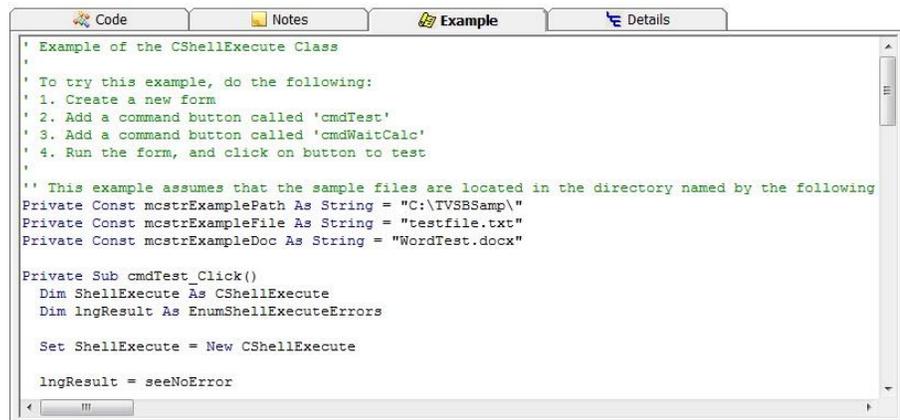
The Notes Tab provides descriptions about the selected item. If you've selected a module/class, it displays the list of properties and procedures:



Notes Tab

Example Tab

The Example Tab provides an example for using the current class or module (examples are not specific to each procedure):

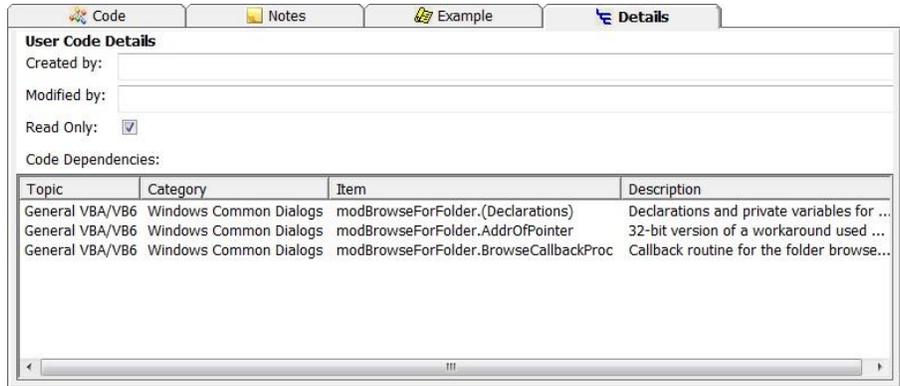


Example Tab

This tab is empty if you are on a topic or category. For more information on using the Example code, see page 28.

Details Tab

The Details Tab shows additional information about the selected item, including code dependencies, and the created/modified user for user-created code:



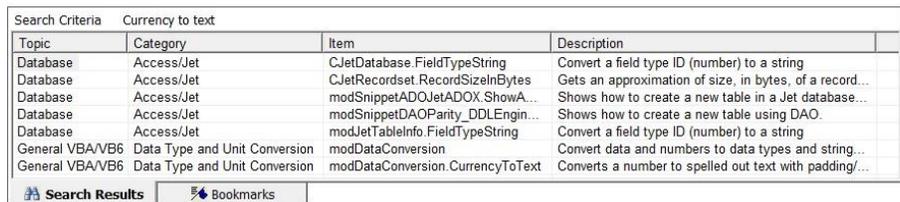
Details Tab

Tracking Window

The Tracking Window appears at the bottom of the Code Explorer window and shows Search Results and Bookmarks.

Search Results Tab

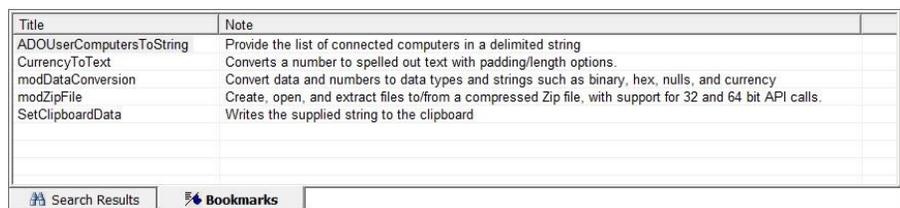
The Search Results tab displays the results from the last Search run. For more information, see **Finding Code with the Search Tool** on page 47.



Search Results Tab

Bookmarks Tab

The Bookmarks tab displays the bookmarks set on code. For more information, see **Creating and Using Bookmarks** on page 50.



Bookmarks Tab

Getting Code into Your Project

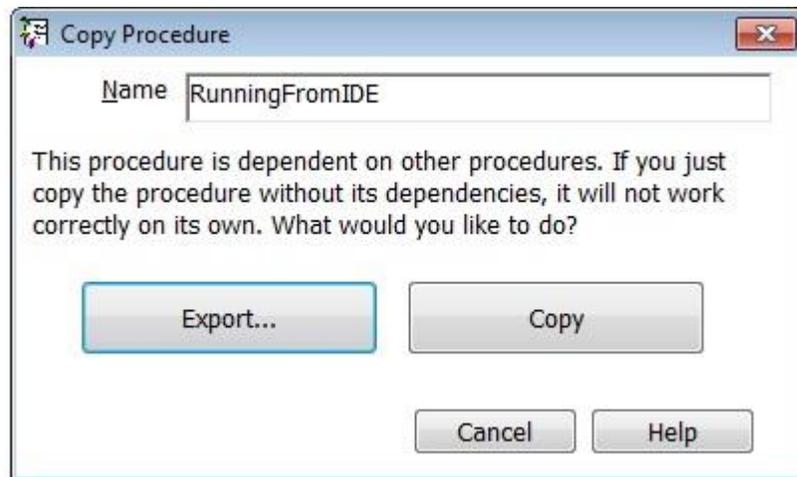
Once you find the class, module, or procedure you want, you need to get it into your project. There are two ways to do this:

- **Copy**—Copy an item to the clipboard, and paste it into your project.

- **Export**—Retrieve an item and its dependencies, and send it to the clipboard, file, or Notepad. If you launched Total Visual SourceBook as an add-in, you can export the code directly into your project.

Copying Code to the Windows Clipboard

The Copy toolbar button  is available to copy the current item to the clipboard quickly. This feature is integrated with the [Ctrl+C] key combination. When an item is displayed in the Code Window, you can press [Ctrl+C] to copy the entire item to the clipboard without having to highlight the item's text. If a module procedure is selected, and the procedure contains dependencies, a dialog appears asking you to copy just the procedure or open the Export dialog to see the dependencies:



Copy or Export Dialog

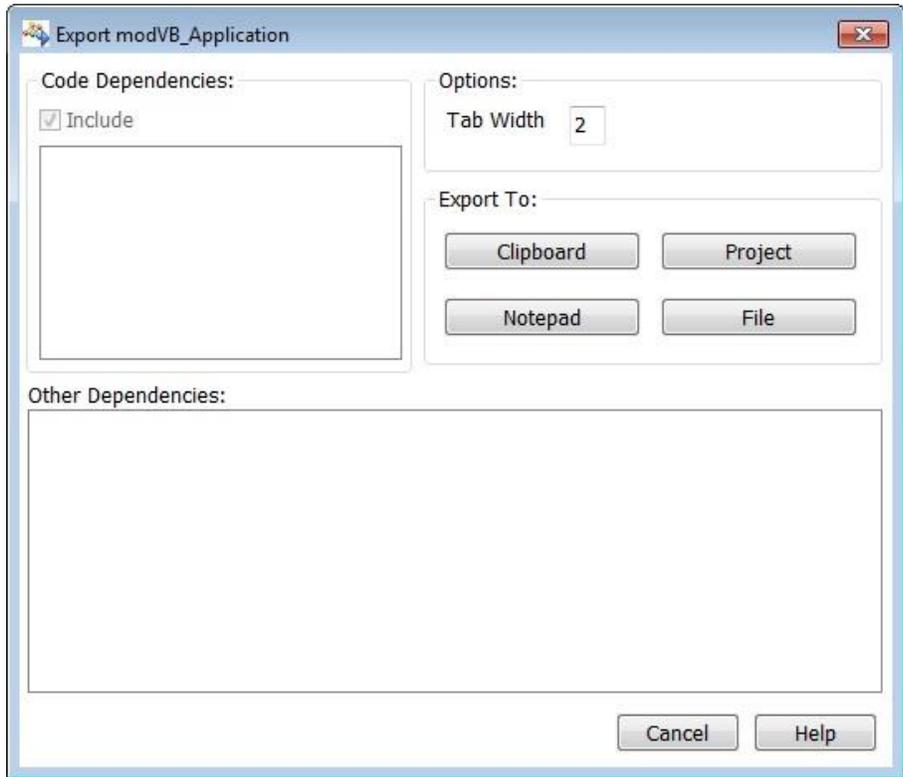


Tip

Note that if you select text and copy text using the Edit, Copy menu item (or the [Ctrl+C] keyboard shortcut), Total Visual SourceBook copies the highlighted text without checking for dependencies.

Exporting Code

Use the toolbar Export button  or select File, Export from the menu:



Export Form

The item name shows in the form's title bar. If the procedure requires other procedures or declarations, they are listed under "Code Dependencies." If there are other dependencies, such as a DLL reference, this is listed under "Other Dependencies."

You can modify the way the exported code is generated by setting the following options:

- **Include Dependencies**— Export the selected procedure and all dependent procedures.
- **Tab Width**— Specify the number of spaces to indent the code.

Set the default values for these options under Tools, Options on the Export tab. For more information, see

Setting Options on page 54.

Getting the Code into Your Project

There are four destinations for exporting code:

- **Clipboard**— The Windows clipboard
- **Notepad**— The Windows Notepad application (NOTEPAD.EXE)
- **File**— A saved file consisting of ASCII text
- **Project**— A VBA project (this is only available if Total Visual SourceBook is launched as an add-in)

Exporting to the Clipboard

The [Clipboard] option sends code to the Windows clipboard. To add it to your project, paste the clipboard contents into an open module. Move to the end of an existing procedure definition ([Ctrl+End]) and paste the clipboard contents ([Ctrl+V] or Edit, Paste from the menu).

Exporting to Notepad

The [Notepad] option sends the code to a Notepad document. Use the Notepad document as a temporary work area, or save it for later use. This is particularly useful if you have code (global variables, constants, user defined types, and declare statements) that needs to be added to the declarations section of a module.

Exporting to a File

Using the [File] option, the exported code is stored in the text file you specify.

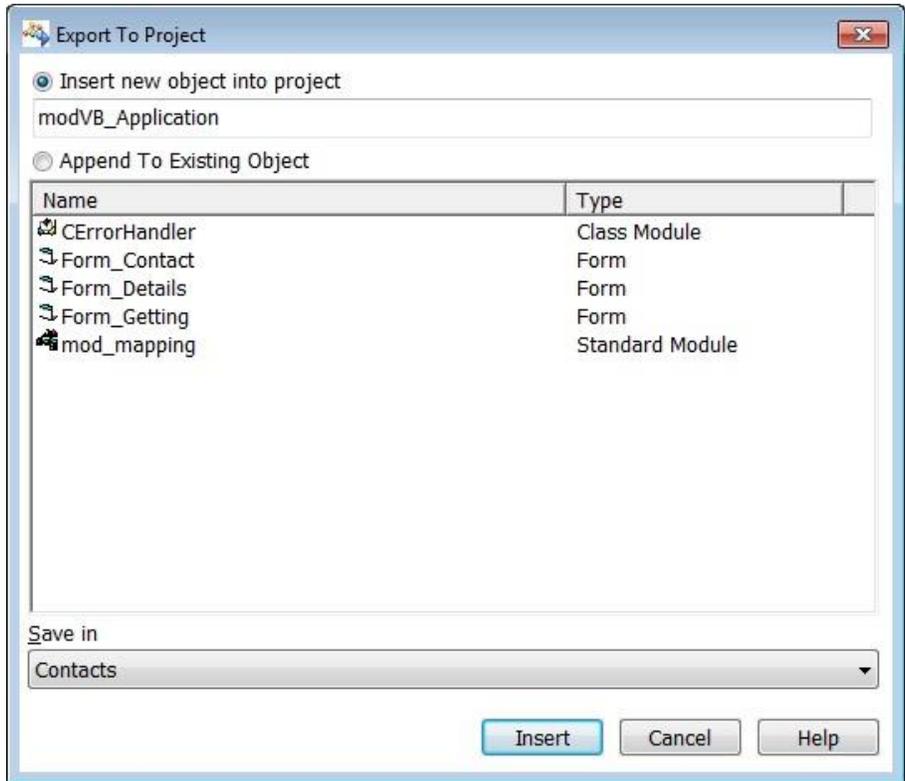
If you are using Visual Basic, you can import the saved file by selecting Project, Add File, and choosing the exported file.

If you are using a VBA host (such as Access or Word), you can import the file by selecting File, Import File, and choosing the exported file.

Exporting to Your Project

The [Project] option exports code directly into your VB/VBA project. There are two options available:

- **Insert New Object into Project**
Create a new object containing the exported code.
- **Append To Existing Object**
Place the exported code into an existing object.
The Append option is only available for modules and procedures.



Export To Project Form

Save in

If you are using Visual Basic and you have more than one Project open, the Save In field allows you to specify the project to export to.

Finding Code with the Search Tool

With the Treeview, much of the code in Total Visual SourceBook can easily be located by category. Still, there may be times that when you have trouble finding code in the Treeview. The Code Explorer's powerful Search Tool helps you quickly locate the code you need based on specified criteria.

Click the toolbar's [Search] button , or select Tools, Search from the menu to open the Search Tool. The Search Dialog is divided into four sections:

- Search Phrase
- Search Category
- Search Options
- Additional Filters for User Code

The code that the Search tool looks through depends on your current view in the Code Explorer. If you are viewing FMS Code only, the Search Tool only searches FMS Code. If you are viewing both FMS Code and User Code, all the code in the Code Group is searched.

Search FMS and User Code

Type the word(s) to search for
OpenModuleCount

Category to search in
<All>

Search Options

Find Whole Words Only

In addition to the item name, search in:

Description Code Example

Additional Filters (This only applies to User Code)

Search in User Notes

Created By: [dropdown]

Modified By: [dropdown]

Creation Date Between 04/14/2013 and 08/14/2013

Modification Date Between 04/14/2013 and 08/14/2013

Search Cancel Help

Search Dialog

Specifying the Search Phrase

In the drop down combo box at the top of the form, enter the word or phrase to find. This can be a single word, such as “table,” or it can be a phrase, such as “days in a month.” The capitalization of your search words does not matter; the search engine performs case-insensitive searches.

Total Visual SourceBook maintains a history of the words and phrases for which you’ve searched. You can choose from a list of previously searched criteria from the combo box.

Boolean Operators

You can include Boolean operators such as AND, OR, and NOT. For example, to find each match with the word “day” and the word “month”, the following search string would be entered “day AND month”. The search phrase can also include parentheses. For example, “day AND (month OR year)” has a different meaning than “(day AND month) OR year”.

Quotes around Phrases

If you want to search for multi-word phrases, use quotes around them. For example, if you type the phrase Jet Engine without quotes around it, AND is assumed and all the items with the word “Jet” and “Engine” are retrieved.

Selecting the Topic and Category to Search

You can specify to search against all the topics and categories or just a specific one. Select the category desired from the combo box.

Setting Search Options

There are several options available in the Search Dialog that allow you to fine-tune how the search operates.

Find Whole Words Only

If this option is selected, only matching words are located.

In Addition to the Item Name, Search in

The item name is always searched for a match. These additional items can also be searched:

- **Description**—The summary text for each item. This is the one line description at the beginning of the Notes pane.
- **Code**—All the text in the code pane, including comments.
- **Example**—All the text in the Example pane.

Specifying Additional Filters for User Code

In addition to the standard search options, there are additional options for searching User Code.

Search in User Notes

Include or exclude User Notes from the search.

Created by

Search by the person who added the procedure. Select the name from the drop down list box.

Modified by

Search by the name of the last user who modified the procedure. Select the name from the drop down list box.

Creation date

Search by the date the user-defined procedure was created.

Modification date

Search by the date the user-defined procedure was last modified.



Because these additional options apply only to User Code, they are disabled when you search while the Code Explorer is only showing FMS Code. To search user code, change the Code Explorer view to display user code or both FMS and user code.

Using the Search Results

After specifying the criteria, click [Search] to execute the search. The results are displayed in the Search Results tab. The top of the tab displays the search criteria you specified, and the matching items are listed below.

Search Criteria	Category	Item	Description
Database	Access/Jet	CJetDatabase.FieldTypeString	Convert a field type ID (number) to a string
Database	Access/Jet	CJetRecordset.RecordSizeInBytes	Gets an approximation of size, in bytes, of a record...
Database	Access/Jet	modSnippetADOJetADOX.ShowA...	Shows how to create a new table in a Jet database...
Database	Access/Jet	modSnippetDAOParity_DDLEngin...	Shows how to create a new table using DAO.
Database	Access/Jet	modJetTableInfo.FieldTypeString	Convert a field type ID (number) to a string
General VBA/VB6	Data Type and Unit Conversion	modDataConversion	Convert data and numbers to data types and string...
General VBA/VB6	Data Type and Unit Conversion	modDataConversion.CurrencyToText	Converts a number to spelled out text with padding/...

Search Results Bookmarks

Search Results Tab

Click on an item in the list and the Treeview jumps to it. You can also click on any column heading to sort the list.

Creating and Using Bookmarks

Total Visual SourceBook provides a powerful Bookmark feature to let you flag your favorite code for easy retrieval. This eliminates the need to search the 2800+ built-in procedures, along with your own code, for the items you commonly use.

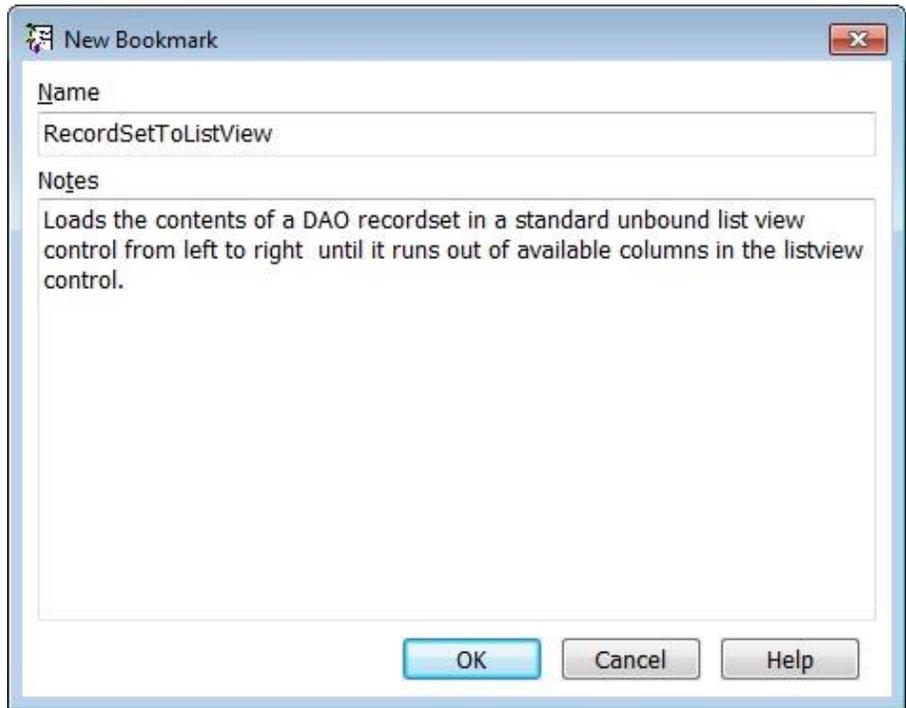
Title	Note
ADOUserComputersToString	Provide the list of connected computers in a delimited string
CurrencyToText	Converts a number to spelled out text with padding/length options.
modDataConversion	Convert data and numbers to data types and strings such as binary, hex, nulls, and currency
modZipFile	Create, open, and extract files to/from a compressed Zip file, with support for 32 and 64 bit API calls.
SetClipboardData	Writes the supplied string to the clipboard

Search Results Bookmarks

Bookmarks Tab

Creating a Bookmark

To add a bookmark, select an item in the treeview and click the New Bookmark toolbar button , or select Edit, New Bookmark from the menu. The New Bookmark Form appears:



New Bookmark Form

Enter a bookmark name (by default it is the selected object's name) and any notes. Then click [OK] to add the Bookmark to your Bookmarks tab.

Navigation and Display

While viewing the Bookmark Tab, you can use the arrow keys and scroll bars to navigate through the bookmarks.

Jump to the Bookmark's Code

Click on a bookmark and the Treeview jumps to it.

Adjust the Bookmark List and Display

Use the mouse to adjust the column widths. Clicking on the column heading sorts the bookmarks on that field.

Right Click Popup Menu

Right-click the bookmark list for a context sensitive menu:



Bookmarks popup menu

Editing a Bookmark

This option allows you to edit a bookmark's name or description.

Deleting a Bookmark

This option deletes the selected bookmark.

Multi-User Issues

Bookmarks are saved based on user name. In a shared environment, you only see your bookmarks. For information on how to re-assign bookmarks, see the **User Name Replace** section on page 65.

Chapter 5: Administrative Tools and Options

Total Visual SourceBook offers a wide range of options to customize the way the code is generated and how the product behaves. There are also administrative features. This chapter covers what you can perform and set from the Tools menu.

Topics in this Chapter

- **Setting Options**
- **Code Databases Tab**
- **Error Handling Tab**
- **Export Tab**
- **Settings Tab**
- **Syntax Highlighting Tab**
- **Administrative Features**

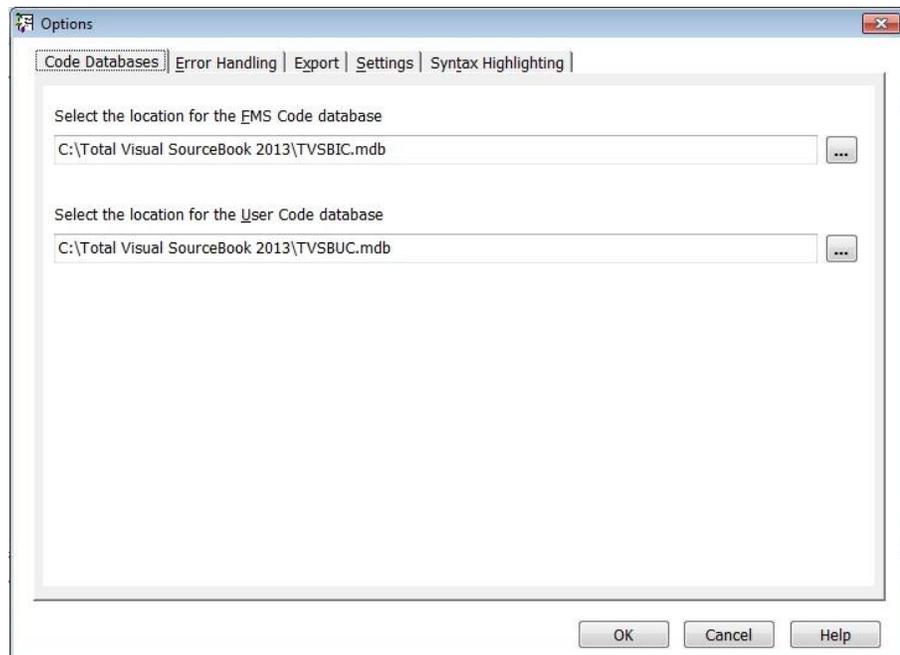
Setting Options

You can specify a variety of settings to control how the Code Explorer behaves, how exported functions are formatted, where the code databases are stored, and your custom error handling code. To set options, select Tools, Options from the menu:



Tools, Options Menu

A tabbed form appears with the options:



Options Form

You can set options for:

- Code Database Locations
- Custom Error Handling
- Exporting Code
- Code Explorer Settings
- Syntax Highlighting Colors and Font

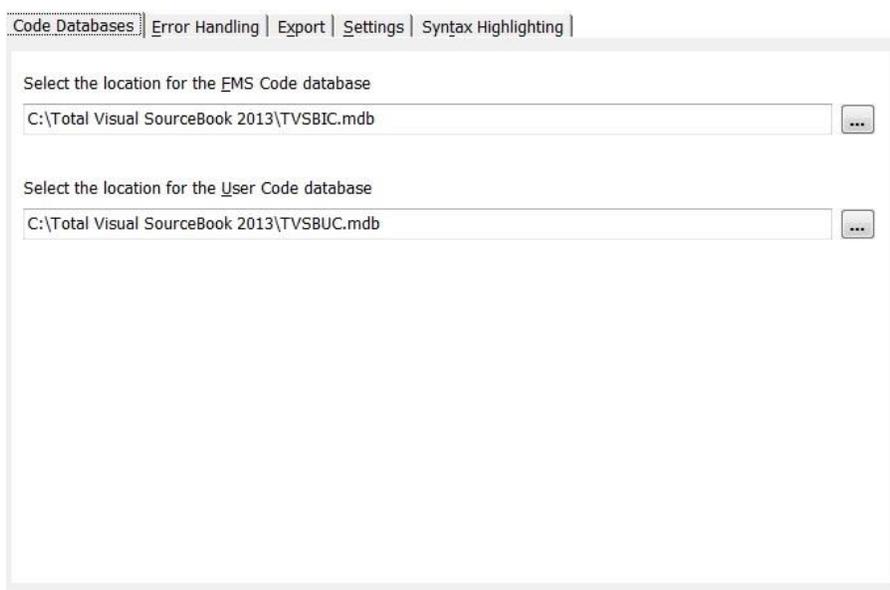
Where Settings are Stored

The settings you specify, with the exception of your name and code database locations, are stored in the shared user code database. This lets you easily set common values for your entire team.

Code Databases Tab

Total Visual SourceBook allows you to place the databases containing the internal and user code in any location on your local or shared network drive. By default, these files are located in the folder where Total Visual SourceBook is installed.

You can specify the file names from the Code Databases tab of the options dialog:



Code Databases Tab of the Options Form

FMS Code Database

The location of the database containing the built-in FMS code (TVSBIC.MDB).

User Code Database

The location of the database containing the code you or your team members have added (TVSBUC.MDB).

If you specify a User Code database that doesn't exist, you are prompted to create a new database there. If you want to move your existing database to

a new location, copy it after exiting Total Visual SourceBook (and any programs that may have it loaded as an add-in), then point to it here.

If you select a file that is shared by other Total Visual SourceBook developers, you must have a licensed copy of the program installed with a unique serial number. Two developers using the same serial number cannot join the same code database.



Note that you can only join a code database that was created by this version of Total Visual SourceBook. You cannot join a database created by an earlier or later version.

Error Handling Tab

In general, every non-trivial subroutine, function, and property in your application should include error handling. Total Visual SourceBook includes default error handling for every code item, but your application's requirements or programming style may require error handling that's different from the default. The Error Handling tab lets you easily replace the default error handling code with your own. There are three sections for the error handler: Enabler, Re-Enabler, and Handler.

If you are not familiar with error handling in VB6/VBA, read our paper on error handling located here:

<http://www.fmsinc.com/tpapers/vbcode/Debug.asp>

Additional information is also available by pressing the Help key on this tab.

Using Tokens

In addition to regular text, Total Visual SourceBook includes a feature to let you insert runtime values called "Tokens" into your error handler with the #Letter# syntax. When these characters are included in the error handling code, Total Visual SourceBook replaces them with the actual values when the code is written to your project. If your error handler includes the line:

```
MsgBox "Error in procedure: " & "#P#"
```

It is inserted into the procedure "ArrayToString" as:

```
MsgBox "Error in Procedure: " & "ArrayToString"
```

These tokens are available:

Token	Description
#M#	The Module Name
#P#	The Procedure Name

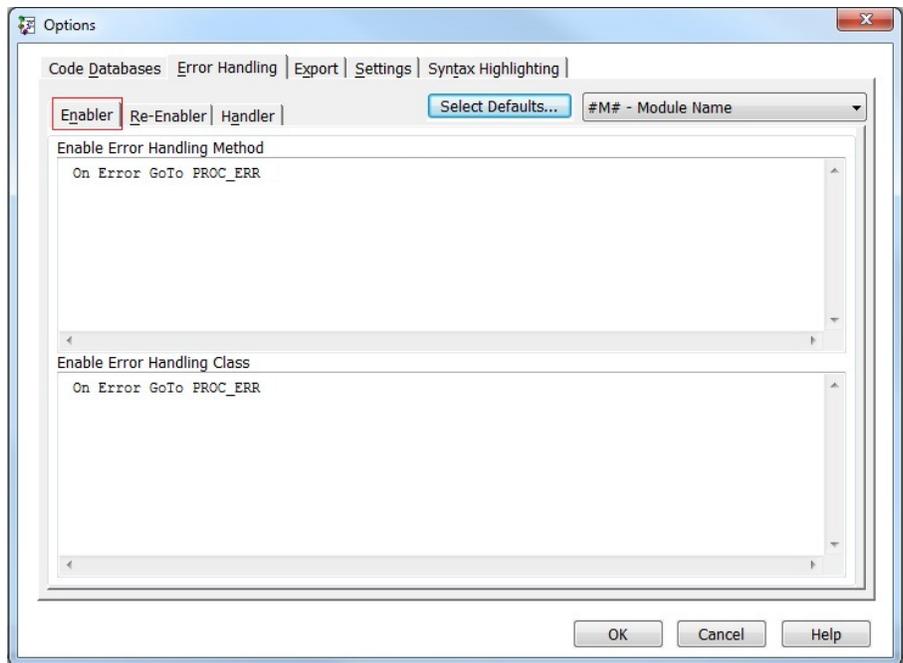
#X#	Exit “Sub”, “Function”, or “Property” (based on the procedure type)
-----	---

Standard vs. Class Modules

You may have different error handling techniques for standard vs. class modules. For instance, you may prefer to have errors in class modules raised back to the caller, rather than having the class members process the error handling with message boxes. Total Visual SourceBook lets you specify different error handling code for standard and class modules.

Error Enabler Tab

The Enabler tab contains the code to enable an error handler:



Enabler Tab on the Error Handling Tab

Use the On Error statement to instruct the application about what to do if it encounters an error. For example:

```
On Error GoTo PROC_ERR
```

tells the program to jump to the PROC_ERR section of the procedure if an error occurs, where it’s hopefully handled gracefully for your users. This line is inserted in every procedure before the first error could potentially occur.

Disabling Error Handling

During development and testing, it’s usually more convenient for the error handling to be disabled so that if a crash occurs, the debugger stops on the offending line. For this feature, we use the gcfHandleErrors constant to turn

the error handler on and off. For this to work, you need to define this as a Public Constant in one of your modules:

```
' Set this to False during development:  
Public Const gcfHandleErrors As Boolean = True
```

Set it to True when you go to production and False during development.

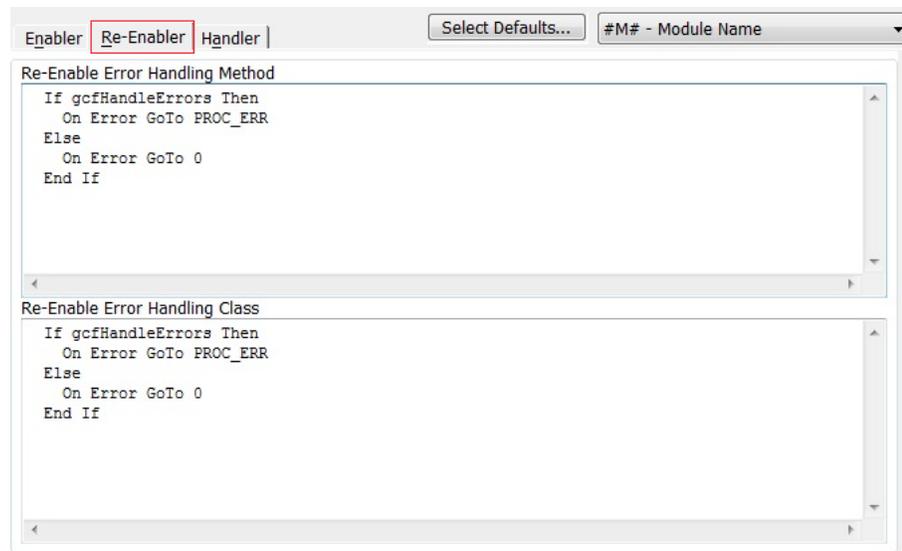
We can then use this as our error enabler:

```
If gcfHandleErrors Then On Error GoTo PROC_ERR
```

To choose this technique, use the [Select Defaults...] button and choose the "Simple" option.

Error Re-Enabler Tab

The Re-Enabler tab includes the code to resume error handling after an On Error Resume Next statement which is handling an expected error such as testing whether a file exists. After the test is completed, error handling should be re-enabled in case an unexpected error is encountered:



Re-Enabler Tab on the Error Handling Tab (with the Simple default selected)

The Simple setting uses the gcfHandleErrors constant to turn error handling on or off depending on whether you're in production mode or not:

```
If gcfHandleErrors Then  
    On Error GoTo PROC_ERR  
Else  
    On Error GoTo 0  
End If
```

Error Handler Tab

The Handler tab includes the error handling code:



Handler Tab on the Error Handling Tab

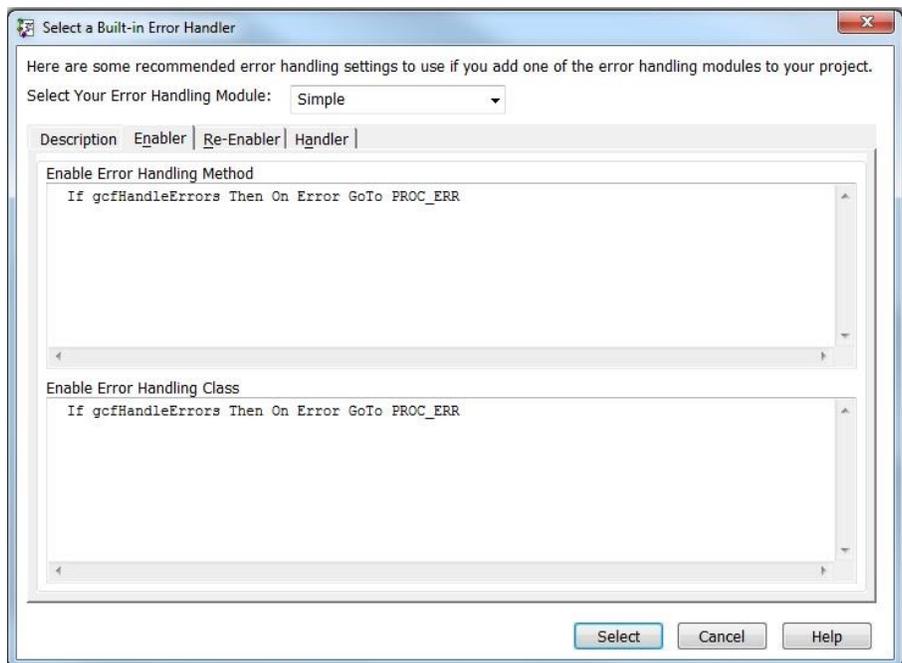
At a minimum, the error handling code should identify the name of the procedure where the error occurs, and display the error message.

Select Default Error Handling Settings

To simplify the specification of the error handling settings, use the

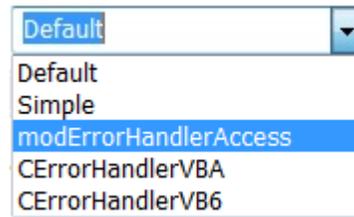


[Select Defaults...] button to choose from predefined settings for the Error Enabler, Re-Enabler, and Handler:



Select Error Handling Settings

Total Visual SourceBook includes several error handling modules and classes you can add to your projects. For each of these, a corresponding set of error routines are available. Choose the one you are using:



Select Your Error Handling Module

When selected from the combo box, you can see its description and corresponding values. Be sure to read the Description tab to understand the settings. You also need to review the examples associated with the error handling module you selected to ensure it is properly initialized when your application starts.

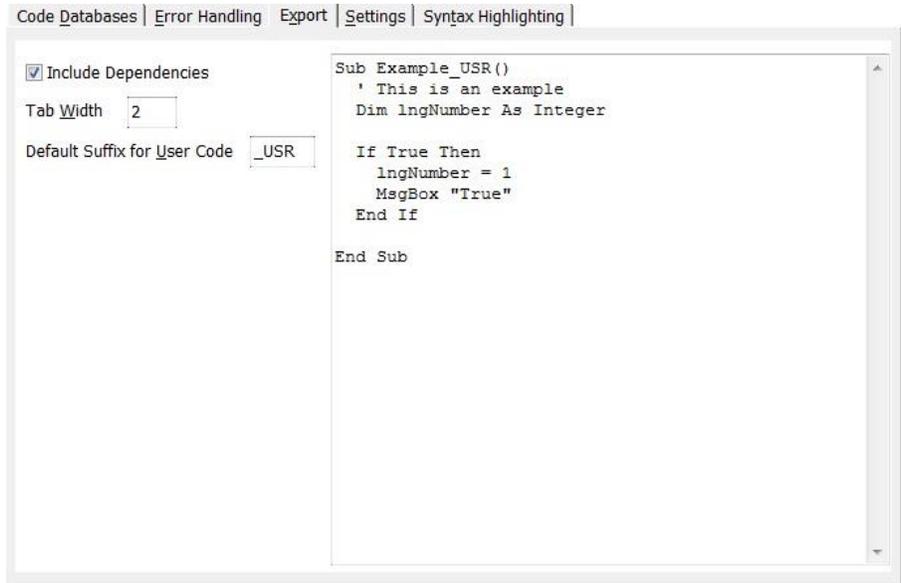
You cannot modify the values here. Press the [Select] button to make it your new error handling values and return to the main error handling page. From there you can customize these default values for your application.

No Error Handling Module

If you are not using a global error handler, choose Default or Simple. Simple includes the use of a global constant to let you easily turn error handling on and off to simplify development/testing versus production. Default omits this feature.

Export Tab

The Export tab includes options for how code is formatted when it is exported. The window on the right displays a preview of your selections.



Export Tab of the Options Form

Include Dependencies

In some cases, procedures require other code in order to work correctly. When you export code, Total Visual SourceBook determines if the code is dependent on any other procedures, and shows you this information in the Export Dialog.

Normally, you should include dependencies, but if you want to manage dependencies on your own, un-check the [Include Dependencies] check box. This option controls the default value on the Export dialog, but you can always override the setting each time you export.

Tab Width

Specify the number of spaces for each indentation level. The default value is two. This setting does not affect the code you add, only the FMS code.

Default Suffix for User Code

Specify the default suffix to use when performing a Save As operation, or clear this field if you don't want a default extension.

Settings Tab

Code Databases | Error Handling | Export | **Settings** | Syntax Highlighting

Code Explorer Settings

- Save Code Explorer Window size on close
- Save procedure position on close

Fonts

TreeView: Arial ...

User Notes: Arial ...

Search Results: Arial ...

Bookmarks: Arial ...

User Settings

- Automatically update Notes with name and date
- User Code is Read Only by default

User Name
Madhuja

TreeView Options

Style
Tree lines, Plus/Minus, Picture, and Text

Settings Tab of the Options Form

Save Code Explorer Window Size on Close

Check this option to have Total Visual SourceBook remember its size and position, and restore it when you reopen the program.

Save Procedure Position on Close

Check this option to have Total Visual SourceBook remember the last procedure displayed.

Fonts

You can customize the fonts for various areas in the Code Explorer:

- **Treeview** – Main Code Explorer Treeview
- **User Notes** – User Notes (also used when printing)
- **Search Results** – List of search results
- **Bookmarks**– List of bookmarks

Note that you can also set the font for the Code display. This option is available on the Syntax Highlighting tab of the Options dialog (see page 55).

Automatically Update Notes with Name and Date

Check this option to automatically update the Notes field each time an item is created or edited. This setting specifies the default behavior, which can be overridden when editing the code.

User Code is Read Only by Default

Check this option to set user code to Read Only by default. Like other settings, this default can be overridden on a case-by-case basis. For more information, see **Marking Code as Read Only** on page 70.

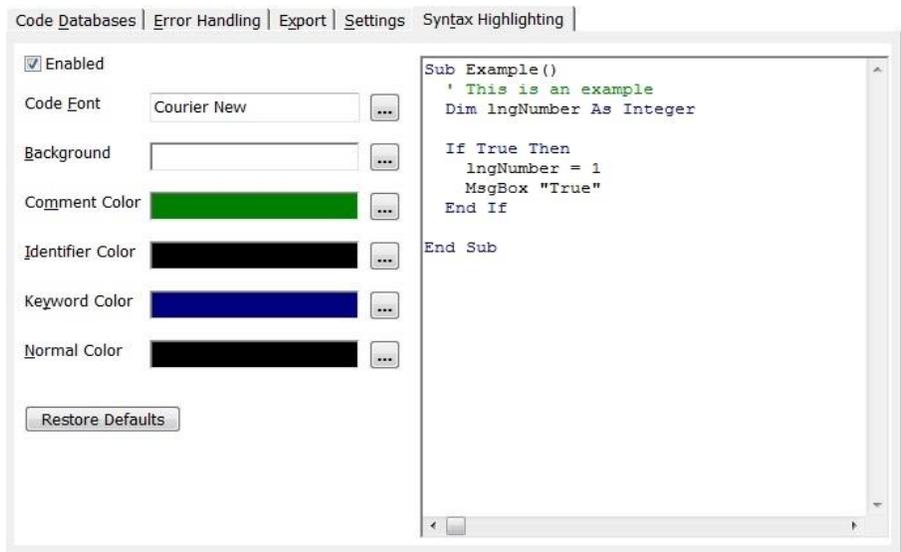
Treeview Options

Specify how the category and procedure information is displayed in the Treeview:

- Text Only
- Picture and Text
- Plus/Minus and Text
- Plus/Minus, Picture, and Text
- Tree lines And Text
- Tree lines, Picture, and Text
- Tree lines, Plus/Minus And Text
- Tree lines, Plus/Minus, Picture, and Text (default)

Syntax Highlighting Tab

Use the Syntax Highlighting options to customize the display of code. The text box on the right displays a preview of your selections. These selections are also used for printing.



Syntax Highlighting Tab of the Options Form

Enabled

Check this option to turn syntax highlighting on or off. If this option is not checked, syntax highlighting is disabled, but background and font selections are still applied to your code.

Code Font

Specify the font used to display code. The default value is Courier New, 10 point. For readability, use a monospaced font, since proportional fonts make indentation and alignment difficult to see.

Background

Specify the background color used to display code.

Comment Color

Specify the color for comments.

Identifier Color

Specify the color for identifiers (procedure names, variables, and constants).

Keyword Color

Specify the color for VB6/VBA keywords.

Normal Color

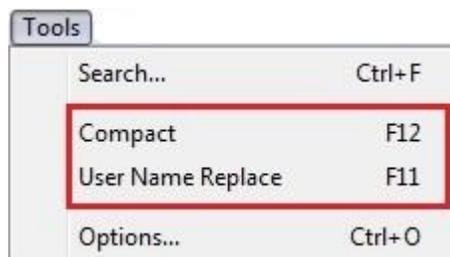
Specify the color for normal characters (words that are not identifiers, comments, or keywords). These include numbers, strings, and mathematical symbols.

Restore Defaults

Restore the default values used by Total Visual SourceBook.

Administrative Features

Total Visual SourceBook offers several functions that make it easy to perform standard administrative functions from the Tools menu:



Administrative Features on the Tools Menu

Compact Database

Total Visual SourceBook stores its code and other configuration information in Microsoft Access/Jet Engine database files. Just like other Access/Jet databases, these need to be periodically repaired and compacted to maintain integrity and efficiency. You should run the Compact process if you add or delete a large number of user-written procedures.

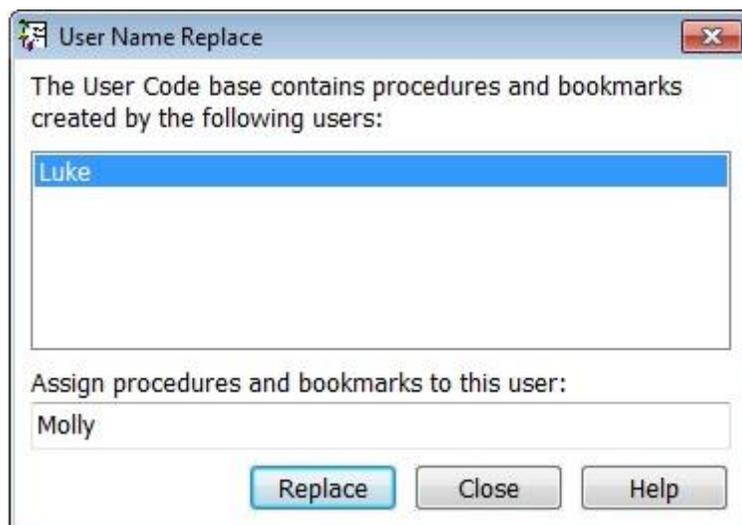


You should always perform a backup of your databases prior to compacting and/or repairing them. You can do this simply by copying the databases to another location, or by ensuring that the files are part of a daily backup routine. You may also want to consider our **Total Visual Agent** program that automates database repair, compact, and backup on a regular schedule. For more information, visit our web site at www.fmsinc.com.

Select the Tools, Compact menu item in Total Visual SourceBook to begin the compact operation. After confirming that you want to perform the operation, it begins. A status bar shows you the progress of the operation.

User Name Replace

When a new procedure is created, the name of the author is stored with the procedure. If you have multiple developers, you may need to re-assign one name to another. Perhaps the original name was entered in error or the author of the code no longer works on the team. The same concept applies to Bookmarks. Bookmarks are stored by user name. You can select Tools, User Name Replace from the menu to re-assign procedures and bookmarks.



User Name Replace Dialog

The list box on the top displays all the developers who added procedures or bookmarks to the Total Visual SourceBook Code Group. Select the entry that matches the name you wish to change, type the name of the new user in the text box, and press [Update].

Total Visual SourceBook runs a mass update of all user-written procedures and bookmarks to change the name. After making the changes the updated list of users is displayed.

Chapter 6: Code Repository

In addition to the Code Explorer and Code Library explained in the previous chapter, Total Visual SourceBook contains a powerful Code Repository feature. This chapter explains how to use the Code Repository to centralize and manage your own source code.

Topics in this Chapter

- **Code Repository Overview**
- **Importance of Backups**
- **Multi-User Issues**
- **System-Maintained Information**
- **Adding New Items**
- **Importing Code**
- **Saving Classes, Modules, and Procedures**
- **Editing Items**
- **Deleting Code**

Code Repository Overview

The code repository lets you store your own source code in Total Visual SourceBook. The code can be from your own work, information you find, or copies and modifications of the built-in code in Total Visual SourceBook. The purpose of the Code Repository is to make your development efforts easier:

- Stop hunting for code in old projects.
- Know where the latest version of your code is stored.
- Eliminate wasted time rewriting lost code.

By keeping your module code with the Total Visual SourceBook code, you have a single place for all your code. You can:

- Add code to any existing category, class, or module.
- Create your own category, class, or module.
- Add notes to your code.
- Use the Code Explorer to find and view your code.
- Use the Search Tool to locate your code.
- Add bookmarks to your code.
- Easily share code with others using the Source Code Exchange feature to package code for distribution.

The Code Repository is multi-user ready and can be shared among multiple developers. This lets the senior developers share their code among the team and not get pestered by repeated requests for their code. The person adding the code can also designate whether other developers have the right to modify or just view it to prevent the new hire from modifying their code.

Importance of Backups

If you use Total Visual SourceBook to manage your code, you must make regular backups of your code.

To back up your code, use your current backup software to save the User Code database. The database file is named TVSBUC90.MDB. Remember that Total Visual SourceBook allows you to have multiple copies of the User Code database, so be sure to create backup copies of all the databases you use.

If you have multiple User Code databases, you must back up each one. For example, you may have installed Total Visual SourceBook both on your local machine and on a shared drive. In such cases, a User Code database exists on your local machine and on the drive where you installed the product for shared use. This scenario requires you to include both database files (the User Code database on your local machine and the shared User Code database on the shared drive) in your backup plan.

The User Code database file is a Microsoft Jet 3.6 database file, and can be archived with any backup software available for Windows. However, do not attempt to backup the file while other developers are using Total Visual SourceBook. Ensure all developers have closed the program before attempting to make a backup.

The frequency with which you make backups is dictated by the amount of code you store in the Code Repository and how often it is updated. The bottom line is that your code is invaluable. Regardless of the tool you use to manage it, you absolutely **need** to have backups.

Multi-User Issues

Total Visual SourceBook allows you to work in a shared environment where you use the same Code Group as other developers. This multi-user capability makes it easy to share the code developed and maintained by your development team, and allows you to make your code accessible to other developers.

The Code Explorer and Code Repository support multi-user access without resorting to complex locking schemes. In other words, the program does not prevent you from making changes based on record locks, because locks are never placed. The program, however, does notify you if you attempt to save edits to an item that changed between the time you opened the item for editing and when you attempt to save it.

Editing Conflicts

If you select a procedure, and select Edit, the data appears in the Edit Procedure/Member dialog. This data is the data in the shared Code Group database at that moment. You can then make and save your changes. However, if another developer edited and saved the same procedure before you saved yours, you could wipe out their changes without realizing it. When you press the [OK] button, Total Visual SourceBook checks to see if the data changed since you started editing. If it detects a change, a dialog

appears. You can then choose to overwrite the other developer's changes or cancel.

If this situation arises, you should copy and paste your changes to a text file for temporary storage. Then determine who made what changes. More importantly, you need to know why someone else is modifying the code you are working on.

Minimizing Contention Issues

There are ways to minimize such overwrite conditions:

- Mark your code as Read Only. Other developers cannot modify code that is set to Read Only by you.
- Make a copy of the code. While you normally don't want your Code Repository to contain multiple versions of the same code, there are cases where this is beneficial. For example, if you plan to make many changes to an existing class or module, it may be better to leave the original version intact and create a new "enhanced" version. This lets you keep the original version for archival purposes, and to support projects where the original version is in use. Of course, tracking multiple versions of code can be difficult, so be sure to keep detailed comments in the Notes field and the code itself.

Refreshing the Treeview

For performance and usability reasons, the Code Explorer does not automatically refresh the Treeview window as soon as other users make changes. If the Treeview were to update with every change, your display could constantly repaint. Additionally, all other operations in the Code Explorer would slow down as the program re-queried the database.

To see the latest view of items in a shared Code Group, press the Refresh button on the toolbar or press [F5].

Marking Code as Read Only

When you add a new category, class/module, or code into the Code Repository, you are the owner and can decide whether other developers can modify it by specifying its "Read Only" setting. You can specify the Read Only status at the individual procedure level, or at a higher level like the category or class/module. The setting is hierarchical. These are the effects of the Read Only setting at each level:

- **Procedure/Property**
If a procedure/property is set to Read Only, only the person who added it can edit or delete it. If Read Only is not set on the procedure, the rights depend on the Read Only setting of the procedure's class/module.
- **Class/Module**
If a class/module is set to Read Only, only the person who created it can modify its information (name and notes) and its members (properties and procedures). No other developers can add, delete, or edit its members, regardless of the member's Read Only setting. The Read Only setting at the Class/Module level takes precedence over the individual member settings.
- **Category**
If a category is set to Read Only, only the person who created the category can modify its name or notes. That person is also the only person who can modify its contents including classes, modules, and procedures.

Basically, if you are not the owner, you can only edit or delete procedures that are not set to Read Only, and are not in a class/module and category set to Read Only. Similarly, you can only add procedures to categories and classes/modules that are not set to Read Only.

Finally, you can only delete categories or classes/modules containing objects for which you have rights. For instance, if you want to delete an entire category, but one of its modules or procedures is set to Read Only by someone else, the Delete is not allowed and no items are deleted.

The Read Only option is set when the category, class/module, or procedure is added. You can change it later by editing it. For more information on changing ownership, see page 65.

System-Maintained Information

To assist with managing your source code, Total Visual SourceBook automatically tracks the following information:

- **Created By** (author name)
When you add new code, your user name is stored as the code's author. This is based on how you identified yourself to the program when you logged in.
- **Created Date**
The date and time the code was created.

- **Modified By**
The name of the person who modified the code.
- **Modified Date**
The date and time the code was modified.
- **Dependencies**
User dependencies are automatically maintained by Total Visual SourceBook. Any user procedure in a module that has a declarations section is automatically flagged as dependent on the declarations. No dependencies are maintained within classes because a class is defined as a complete unit—its dependencies are implied. For user code, Total Visual SourceBook automatically creates a dependency for modules that have a declarations section. You cannot alter this dependency or add any custom dependencies to user code.

With this information, you can use the Search Tool to locate your procedures. For details, see page 47.

Adding New Items

Adding a New Topic

The highest level of organization in the Total Visual SourceBook is the Topic level. You can add code to an existing topic or create your own. To add a new topic, open the New Topic form by pressing the  [New Topic] button on the toolbar or selecting File, New Topic from the menu:

New Topic Form

Name, Description, and Notes

Enter a unique name for the new topic, and type a description. This description should be brief enough to be understood quickly when viewed in the Code Explorer.

The Notes field allows you to include additional documentation for the topic. Notes are stored in rich text format (RTF), and can have a variety of formatting features. Click [Open Editor] to edit the RTF notes in a full-featured RTF editor.



The FMS Rich Text Editor is part of the Total Access Memo program, which is available through FMS, Inc. The built-in FMS Rich Text Editor gives you these features and works like other text editors such as WordPad or Microsoft Word. As such, this manual does not go into detail on its features. Additionally, the Rich Text Editor includes its own help file, which is available when open the Editor, and you press [F1], or choose the Help, Contents menu item.

Other Options

Two additional options allow you to customize your new topic entry:

- **Update Notes with Your Name and Today's Date**
Automatically adds the user name and current date/time to the Notes. This is useful for tracking how items change over time.

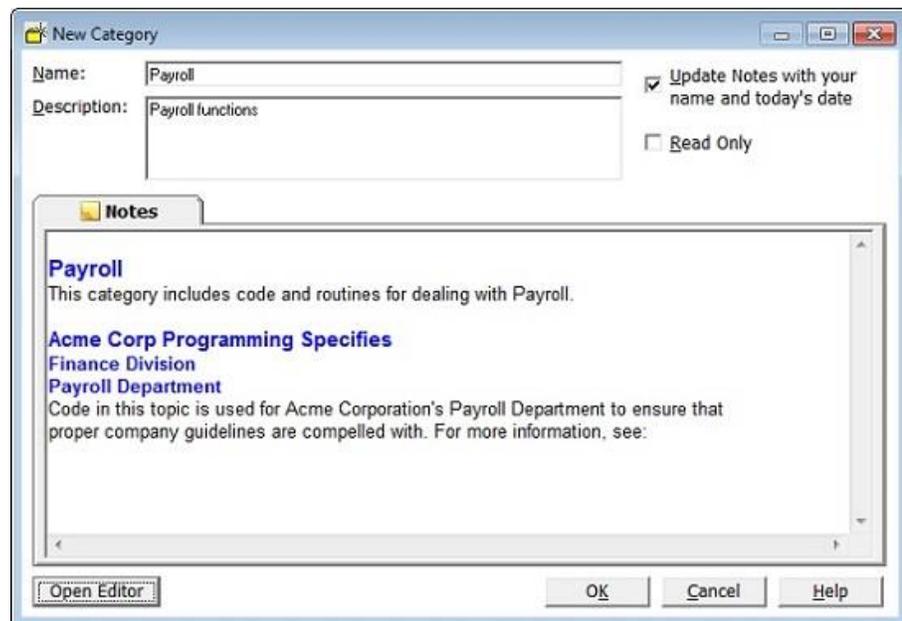
- **Read Only**

Disallows other users from editing or deleting the topic, and only allows the owner of the topic (you) to make changes or delete it. This setting also prevents people from modifying any of the topic's contents.

Defaults for these options are stored under Tools, Options (on the Other tab).

Adding a New Category

You can add code to an existing category or create your own. To add a new category, open the New Category form by pressing the  [New Category] button on the Code Explorer toolbar or by selecting File, New Category from the menu:



New Category Form

Name, Description, and Notes

Enter a unique name for the new topic, and type a description. This description should be brief enough to be understood quickly when viewed in the Code Explorer. This field can be used in the Search Tool, so enter key words or other phrases to make it easier to find.

The Notes field allows you to include additional documentation for the category. Notes are stored in rich text format (RTF), and can have a variety of formatting features. Click [Open Editor] to edit the RTF notes in a full-featured RTF editor. Notes can also be searched, so be sure to include important keywords that will help you find the category later on.



The FMS Rich Text Editor is part of the Total Access Memo program, which is available through FMS, Inc. The built-in FMS Rich Text Editor gives you these features and works like other text editors such as WordPad or Microsoft Word. As such, this manual does not go into detail on its features. Additionally, the Rich Text Editor includes its own help file, which is available when open the Editor, and you press [F1], or choose the Help, Contents menu item.

Other Options

Two additional options are available for customizing your category:

- **Update Notes with Your Name and Today's Date**
Automatically adds the user name and current date/time to the Notes. This is useful for tracking how items change over time.
- **Read Only**
Disallows other users from editing or deleting the category, and only allows the owner (you) to make changes or delete it. This setting also prevents people from modifying any of the category's contents.

Defaults for these options are stored under Tools, Options (on the Other tab).

Adding a New Class or Module

Each category contains one or more classes, modules, and snippets. Select the category, then click the  [New Class or Module] button on the Code Explorer toolbar or select File, New Class/Module from the menu:

New Class or Module Form

Name

Enter a name for your new module, class, or snippet, keeping in mind these guidelines:

- The name must be unique within the category.
- Although Total Visual SourceBook does not enforce naming rules, we recommend following your programming environment's rules regarding name length and valid characters.
- Total Visual SourceBook uses the convention of prefixing module names with "mod" and class names with "C." For consistency, we suggest using this convention for user code as well.

Module Type

Specify whether the entry is a module, a class, or snippet by selecting the appropriate Module Type option. Total Visual SourceBook uses Snippet modules for code that demonstrates techniques, but can't run on its own. Snippets are similar to standard Modules, but their Treeview icon is different.

Description

Type a brief description for the new item. Since the Search Tool can search on this field, enter keywords or other phrases to make it easier to find.

Notes

The Notes field allows you to include additional documentation for the item. Notes are stored in rich text format (RTF), and can have a variety of formatting features. Click [Open Editor] to edit the RTF notes in a full-featured RTF editor. Notes can also be searched, so be sure to include important keywords that will help you find the category later on.



The FMS Rich Text Editor is part of the Total Access Memo program, which is available through FMS, Inc. The built-in FMS Rich Text Editor gives you these features and works like other text editors such as WordPad or Microsoft Word. As such, this manual does not go into detail on its features. Additionally, the Rich Text Editor includes its own help file, which is available when open the Editor, and you press [F1], or choose the Help, Contents menu item.

Example

Under the Example tab, add code to show how to use the module or class.

Options

The following options let you control the settings of your new module or class:

- **Update Notes with Your Name and Today's Date**

Check this option and Total Visual SourceBook automatically adds information to the Notes with your user name, and the current date and time. This is particularly useful if people change the item information over time.

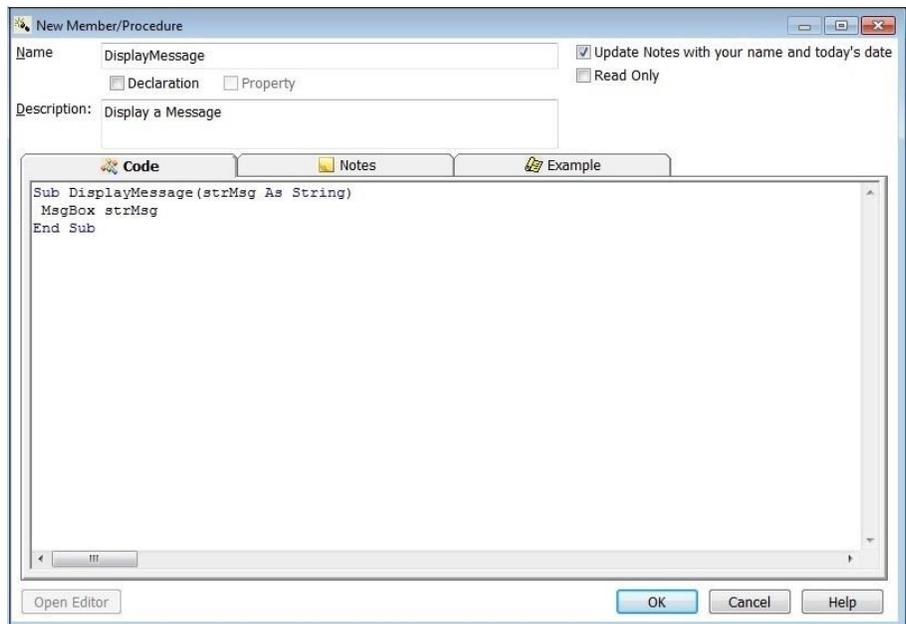
- **Read Only**

Check this option if you do not want others to edit or delete this item. Enabling the Read Only setting only allows you (the owner) to make changes or delete it provided the category and class/module are not set to Read Only by someone else. This prevents new hires from modifying or deleting your code.

Defaults for these options are stored under Tools, Options (on the Other tab).

Adding a New Member or Procedure

To add a new member or procedure to an existing class or module, select the class or module in the Treeview and press the New Member or Procedure button  on the Code Explorer toolbar, or select File, New Member/Procedure from the menu:



New Member or Procedure Form

Name

Enter a name for the item, keeping in mind these guidelines:

- The name must be unique within the class or module.

- Although Total Visual SourceBook does not enforce naming rules, we recommend following your programming environment's rules regarding name length and valid characters.
- The name you assign to a new member or procedure is not tied to the actual code you enter into the Code Window. For example, you could create a procedure named "CloseFile" and then enter a procedure in the Code Window named "PurgeFile". This allows you to have multiple versions of the same procedure name stored under different names in the Code Explorer Treeview.

Declaration

This option mark the code item as a declaration. Total Visual SourceBook automatically creates a dependency for every procedure in the module to the declaration. This option is not available if a procedure in the same class or module is already marked as a declaration.

Property

This option marks the code item as a property. This setting only affects the icon displayed in the Code Explorer Treeview. This option is only available if you are adding a member to a class.

Description

Type a brief description for the new item. Since the Search Tool can search on this field, enter keywords or other phrases to make it easier to find.

Code

Type or paste the actual code for the procedure or class member. Total Visual SourceBook does not validate or compile the text you enter into the Code field. Although it expects you to enter valid code, there is nothing to stop you from pasting in a text file, an email message, or whatever else you may have. However, to use Total Visual SourceBook in the most effective way, put code into the Code text box in the same format that Total Visual SourceBook uses for the built-in code:

- In general, you should store only one procedure's code per procedure/member item. An exception is when you are pasting in a property procedure, in which case you should place all the pieces (Let, Get, Set) for that property.
- Avoid blank lines at the beginning and end of your procedure's code. This makes the Export tool work correctly when pasting the code into your VB project.

- Paste in tested code. You should adopt the practice of never typing into the Code textbox. Instead, write and test the procedure, then copy the code from the VB/VBA editor window, and paste it into the Code textbox. This eliminates the formatting and typing errors that can occur from manually editing.

Notes

The Notes field allows you to include additional documentation for the item. Notes are stored in rich text format (RTF), and can have a variety of formatting features. Click [Open Editor] to edit the RTF notes in a full-featured RTF editor. Notes can also be searched, so be sure to include important keywords that will help you find the category later on.



The FMS Rich Text Editor is part of the Total Access Memo program, which is available through FMS, Inc. The built-in FMS Rich Text Editor gives you these features and works like other text editors such as WordPad or Microsoft Word. As such, this manual does not go into detail on its features. Additionally, the Rich Text Editor includes its own help file, which is available when open the Editor, and you press [F1], or choose the Help, Contents menu item.

Setting Options

The following options allow you to control the settings for your new procedure or class member:

- **Update Notes with Your Name and Today's Date**
Check this option and Total Visual SourceBook automatically adds information to the Notes with your user name, and the current date and time. This is particularly useful if people change the item information over time.
- **Read Only**
Check this option if you do not want others to edit or delete this item. Enabling the Read Only setting only allows you (the owner) to make changes or delete it provided the category and class/module are not set to Read Only by someone else. This prevents new hires from modifying or deleting your code.

Defaults for these options are stored under Tools, Options (on the Other tab).

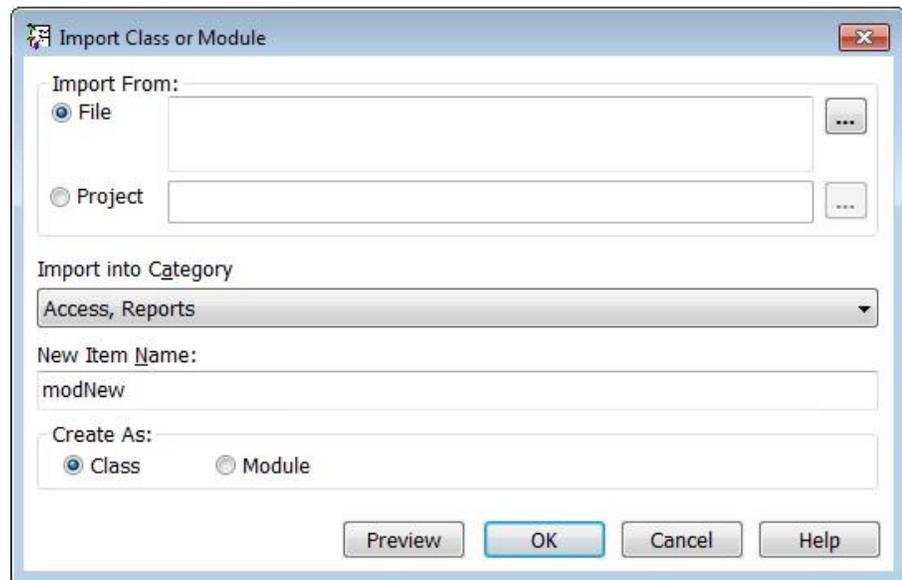
Importing Code

Rather than manually entering each procedure or property of a class or module, you can import from a file or Visual Basic Project directly into the Total Visual SourceBook Repository. In this context, “Visual Basic Project” includes Visual Basic 6.0 and any code modules or classes that exist in Office documents (such as Access databases or Excel workbooks). The import feature imports a whole class or module at once—it does not import individual procedures.



For information on importing code from another source code library program or a Source Code Exchange (SCE) file, see Chapter 7: Source Code Exchange.

To import an item, select File, Import, Text File/Current Project from the menu. The Import Dialog appears:



Import Dialog

Import From

First, specify the source of the item you are importing:

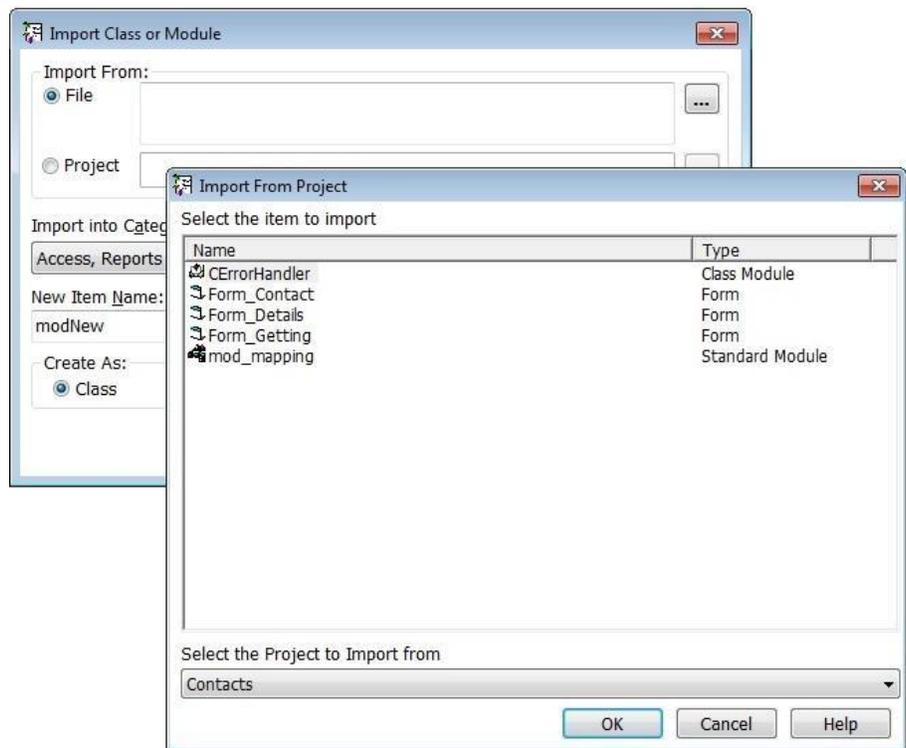
File

This can be any file, but only code information from the file is imported. See **How Non-Code Data Is Imported** on page 81 for more information.

Project

Import a module from your currently open Visual Basic Project. Note that in order to import from a Project, you must have Total Visual SourceBook

open as an add-in through Visual Basic or your Office application. Click [...] to choose the module to import:



Import from Project Form

All Project items that contain code are shown, but only code information from an object is imported. Please see **How Non-Code Data Is Imported** on page 81 below for more information.

Specify the Destination

Next, specify the topic and category to import into, and the name of the item. The name is automatically filled in from the file or object name. Specify the destination type of object: class or module. This does not depend on the type of object you are importing.

How Non-Code Data Is Imported

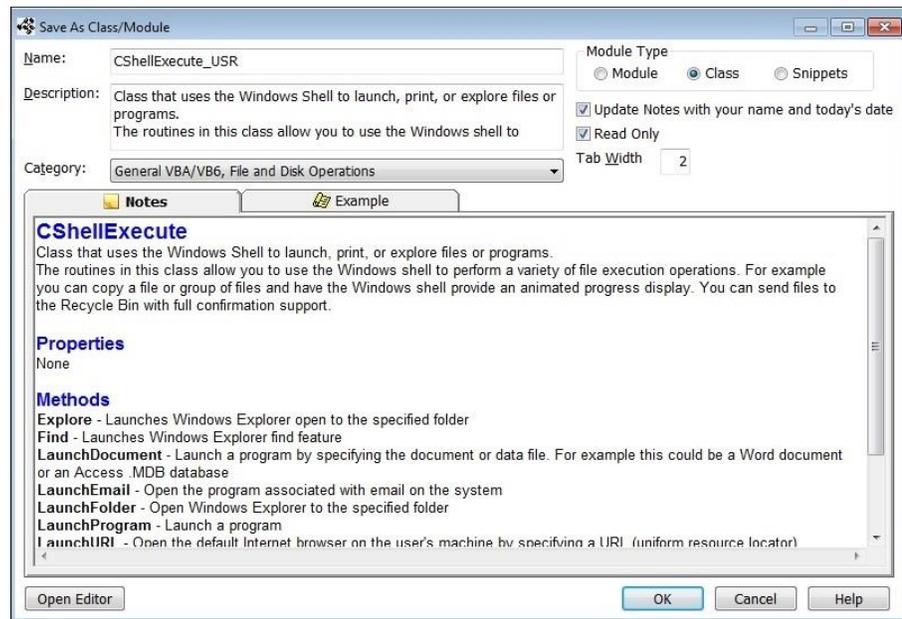
The Import operation does not store any data from a source object that is non-code. Non-code data includes form layout information, designer attributes, and module and class attribute information. This information is discarded by the Import operation. This behavior is important to note if you plan on importing class modules into Total Visual SourceBook and the modules contain extended information about default properties, etc. This information needs to be manually recreated when you export that class back into a VB/VBA project.

Saving Classes, Modules, and Procedures

Save As Class/Module

You can save an existing class or module under a new name in the same category, then edit and change the code to suit your needs. You can do this on any class or module, whether it is FMS code, user code, or even code marked as Read Only.

Select the module or class in the Total Visual SourceBook Code Explorer Treeview. Press the  [Save As] toolbar button or select File, Save As from the Total Visual SourceBook Code Explorer menu. The following dialog appears:



Saving a Class or Module with a New Name

Class or Module Name

By default, the name is the original name with the trailing suffix specified on the Options form (see page 61).

Category to Save In

You can specify the topic and category in which to save the item. By default this is the current category.

Other Options

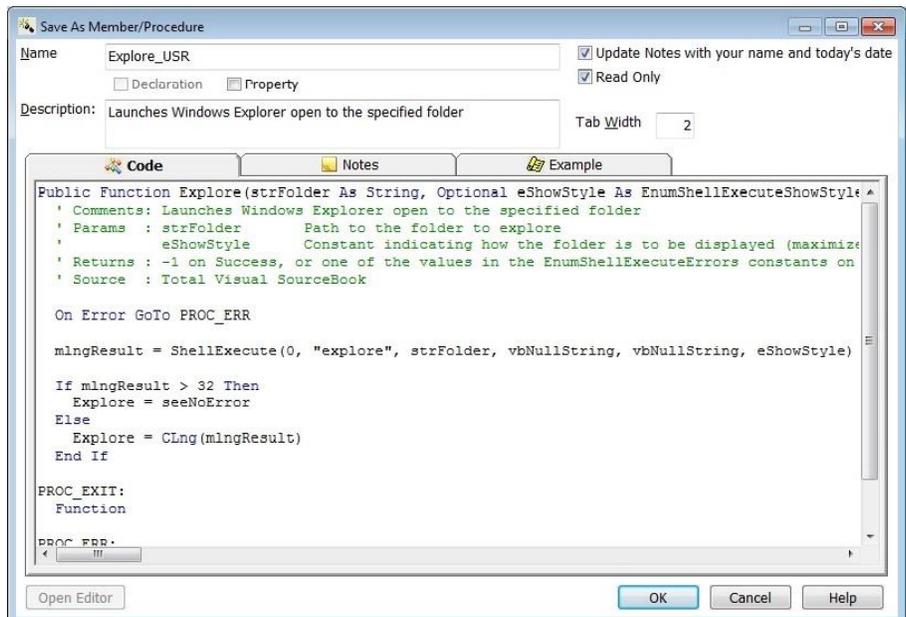
All other options and entries on this dialog are the same as when you create a new class or module. See **Adding a New Class or Module** on page 75 for more information.

Save As Member or Procedure

You can save an existing procedure under a new name in the same class/module, then edit and change the code to suit your needs. Save As can be used on any procedure, whether it is FMS code, user code, or even code marked as Read Only.

Select the procedure in the Total Visual SourceBook Code Explorer

Treeview. Press the  [Save As] toolbar button or select File, Save As from the Total Visual SourceBook Code Explorer menu. The following dialog appears:



Saving a Procedure with a New Name

Procedure Name

By default the suggested name is the original name with a trailing suffix consisting of the value specified on the Options form (see page 61).

The same rules for names and conflicts that applied to creating a new procedure also apply when you use the Save As function. See **Adding a New Member or Procedure** on page 77 for more information.

Tab Width

This option specifies the number of spaces for each indentation level.

Other Options

All other options and entries on this dialog are the same as when you create a new procedure or class member. See page 77 for more information.

Editing Items

You cannot edit built-in FMS topics, categories, and code, but you can modify user-created topics if you created the topic or if it is not designated as Read Only by someone else. If you need to change the ownership of a topic after it is created, use the User Name Replace option (see page 65).

To Edit the currently selected item in the Treeview, choose Edit, Edit from the menu, press [F2], or click the toolbar button . You can edit the following item types:

- **Topic**
The options on this form are the same as those for the New Topic form. See page 72 for more information.
- **Category**
The options on this form are the same as those for the New Category form. See page 74 for more information.
- **Class/Module**
The options are similar to those for the New Class or Module form. See page 75 for more information.
- **Procedure**
The options on this form are the same as those for the New Member or Procedure form, explained on page 77.

Deleting Code

You cannot delete the FMS items, but you can delete user-items, provided that you are the creator or the creator has not marked the items as Read Only. For more information on the Read Only settings, see page 70.

From the Total Visual SourceBook Code Explorer Treeview, select the category, class/module, or procedure to delete and press the  [Delete] button on the toolbar, or select File, Delete from the Code Explorer menu. You may also simply press the [Del] key when the selected item is highlighted. Items cannot be recovered once deleted.

Chapter 7: Source Code Exchange

This chapter describes how you can import data from other code libraries, and use the Source Code Exchange (SCE) tool to share code with other Total Visual SourceBook users. This feature packages code, notes, and examples, and preserves topic and category information for automated import into Total Visual SourceBook.

Topics in this Chapter

- **About Source Code Exchange**
- **Creating an SCE Package**
- **Importing an SCE Package**
- **Importing Data from another Code Library**

About Source Code Exchange

Total Visual SourceBook Source Code Exchange files (SCEs) provide an easy way to share or publish your code. SCEs are packages of code and notes extracted from a Total Visual SourceBook user code database. Because Source Code Exchange handles replicating code and notes, you can use it to keep multiple versions of code in sync.

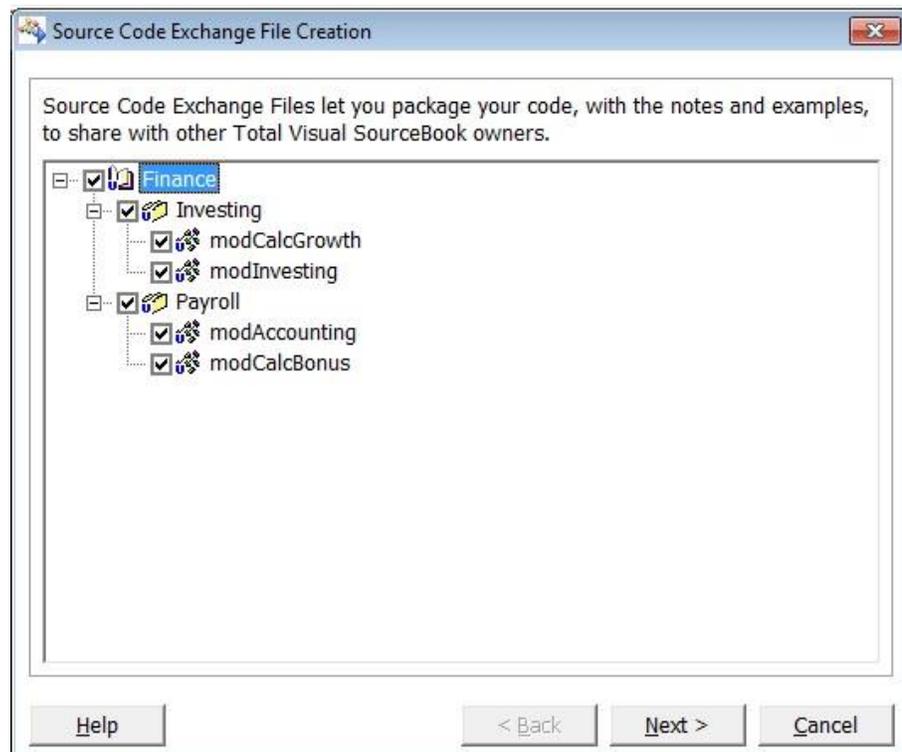
Use SCE files to:

- Send updated code to offsite developers.
- Share new techniques with your programming team.
- Create subject-matter specific code for easy inclusion in a project.
- Post your own code on the web.

In addition, Total Visual SourceBook code library updates are provided in the form of SCE files that you can obtain from our product support web site.

Creating an SCE Package

To create an SCE file, select File, Create SCE File from the menu. This opens the first page of the SCE File Creation Wizard:



SCE Export Form

All of the available user code is shown in the treeview list. Select the items to export and click Next.



You can only select an entire module or class. It is not possible to export only certain procedures or properties within a module or class. Only user-written modules can be exported, but both user and FMS-written topics and categories can be exported. The SCE file always contains the complete hierarchy of storage from Topic to Category to Module. This means that you can always import this SCE file into another copy of Total Visual SourceBook, even if the other copy does not have the same topics and categories as your system.

On the second page of the Wizard, choose where to save the SCE file, and enter additional details about the file:

Source Code Exchange File Creation

Save As:
C:\Total Visual SourceBook 2013\Export.sce

Developer (Required):
Molly

Email (Required):
fms@fmsinc.com

Description:
Acme corporation financial code routines.

Copyright Information:
Copyright[c]Acme, Inc.This code is for use only within Acme Corporation.It may not be distributed without express written permission.

Help < Back Finish Cancel

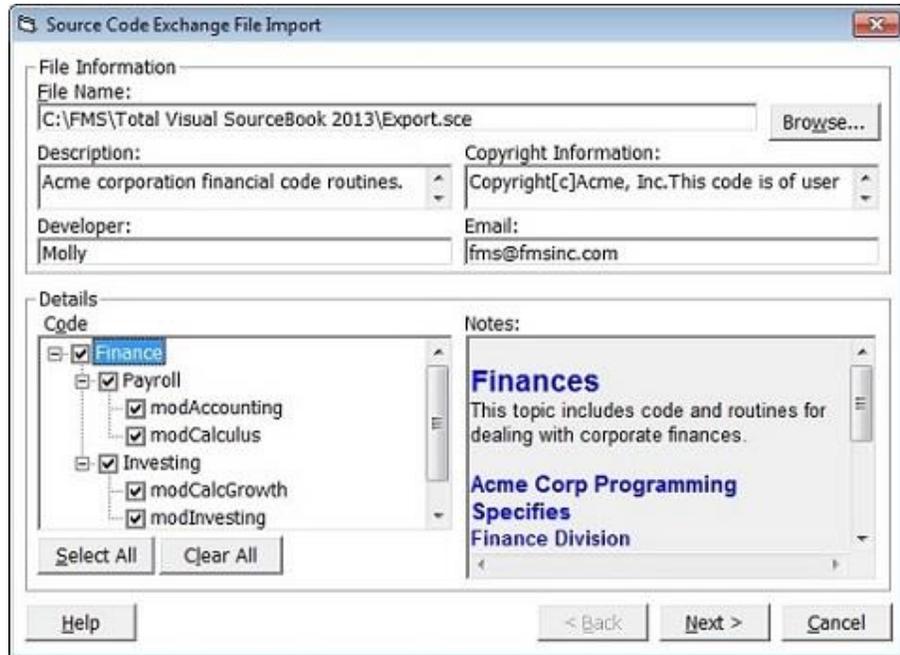
SCE Export Descriptive Fields

The additional details—Developer, Email, Copyright Information, and Description—is useful for finding the SCE file to import at a later time.

After entering the details, click [Finish] to create the file.

Importing an SCE Package

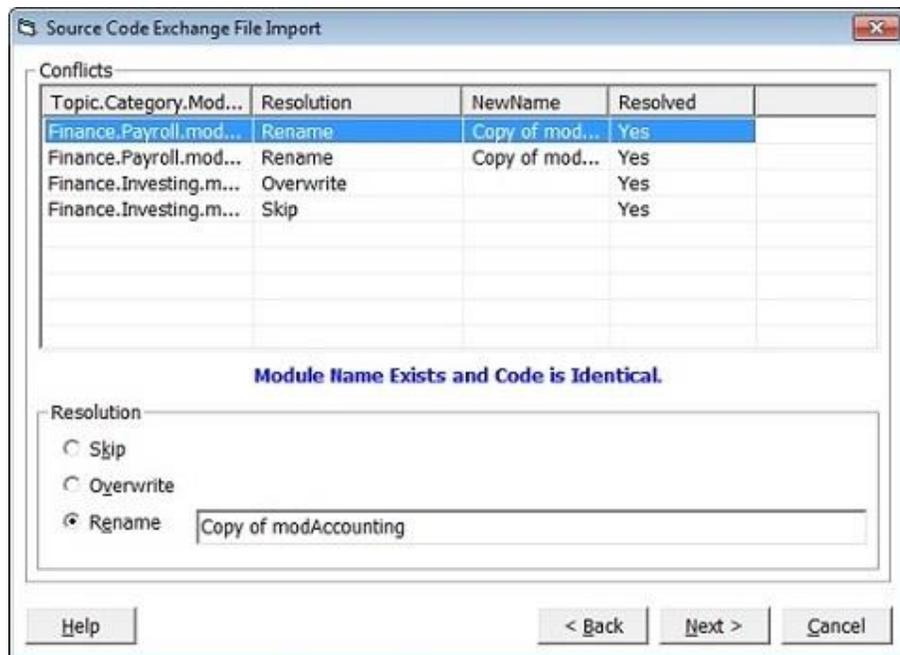
To import a Source Code Exchange (SCE) from another user, select File, Load SCE File to display the SCE import dialog:



SCE File Import Dialog

Click [Browse] to select the SCE file to import, and the details are displayed on the form. The code in the SCE file is displayed in a hierarchical Treeview—you can import all code, or select only certain items to import. Use the [Select All] button to select all items in the list, and the [Clear All] button to unselect all items.

After selecting the desired code, click [Next] to display the Conflicts form:



SCE File Import Conflicts

There may be conflicts between the names of the code you are importing and the items currently in the database. A list of possible conflicts is displayed in the lower portion of the screen.

To help Total Visual SourceBook determine how to resolve these naming conflicts, you must select one of the following options for each conflicting module:

- **Skip**—Do not import the item.
- **Overwrite**—Replace the existing item in the database with the new item without making a backup.
- **Rename**—Select a new name for the imported item.

After resolving all naming conflicts with the SCE file, click [Next] to continue, then click [Import] to begin the import.

Importing Data from another Code Library

Total Visual SourceBook allows you to import data from the following code library programs:

- Total Access SourceBook 2.0 for Microsoft Access 2.0
- Total Access SourceBook 8.0 for Microsoft Access 95 and 97
- Total VB SourceBook 5 for VB 5
- Total VB SourceBook 6 for VB 5 and 6
- Microsoft Office Developer Code Librarian

Special Note for Microsoft Office Developer Code Librarian

The MOD Code Librarian stores its data in a password-protected Access 2000 (Jet 4.0) database. Because Total Visual SourceBook uses version 3.6 of the Microsoft Jet database, our program cannot directly open this file. To import data from the MOD Code Librarian, follow these steps before you attempt to import the data into Total Visual SourceBook.



Total Visual SourceBook uses Jet 3.6 because it is sufficient for the functionality required, uses fewer files, and is much smaller than a similarly deployed Jet 4.0 application.

Step 1: Start Access

Open Access, version 2000 or later. If you do not have Access installed on the machine you are working on, you can follow these steps on another machine.

Step 2: Locate your MOD Code Librarian database

Find the CodeLib.mdb file that contains your MOD code. By default, this file is located in the folder where you installed the Microsoft Office Developer tools, e.g. C:\Program Files\Microsoft Office\ODETools\V9

Step 3: Open the MOD Code Librarian

In Access 2000 or later, select File, Open and point to the database you located in Step 2 above. When prompted for a password, enter:

CodeLib

Note that the password is case-sensitive.

Step 4: Convert the database to a compatible format

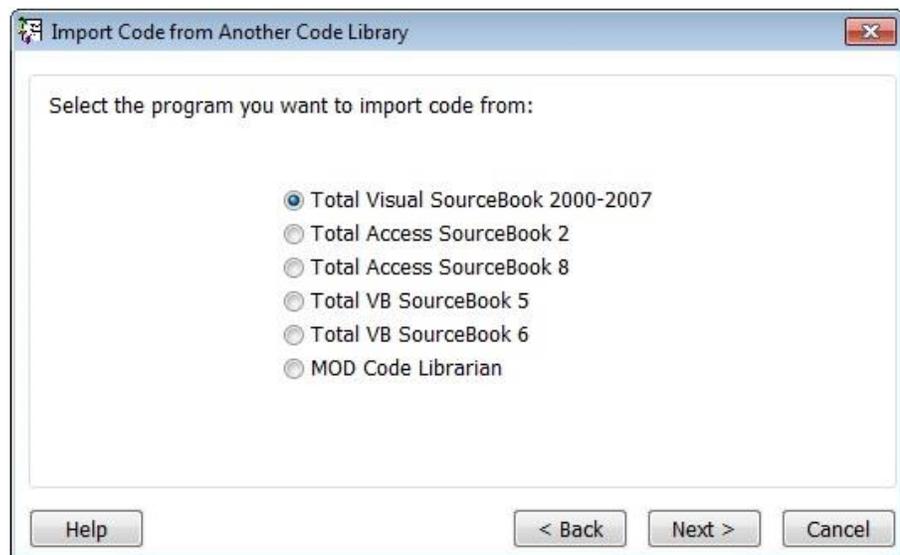
Once the database is open, select Tools, Database Utilities, Convert Database, To Prior Access Database Version from the Access menu. The Access Save File dialog appears asking you where you want to save the converted copy of the database.

Step 5: Finish the conversion and close Access

After saving the new file, close Access and return to Total Visual SourceBook, and begin the import.

Starting the Import Process

To import a file, select File, Import, Another Code Library from the menu to open the first page of the Import Code wizard:



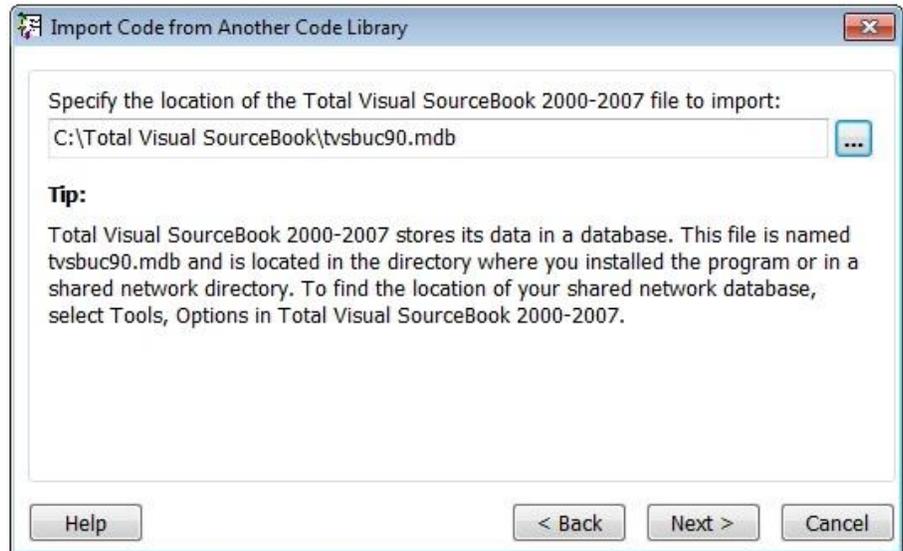
Import Code from Another Code Library

Select the product from which you wish to import and click [Next] to continue.



When importing from Total Access SourceBook (version 2.0 or 8.0), you must have Total Access SourceBook installed on the local computer. Because of the structure of the data in Total Access SourceBook 2.0 and 8.0, importing from a shared drive across a network will not work.

On the second page of the Import Wizard, select the location of the file to import:

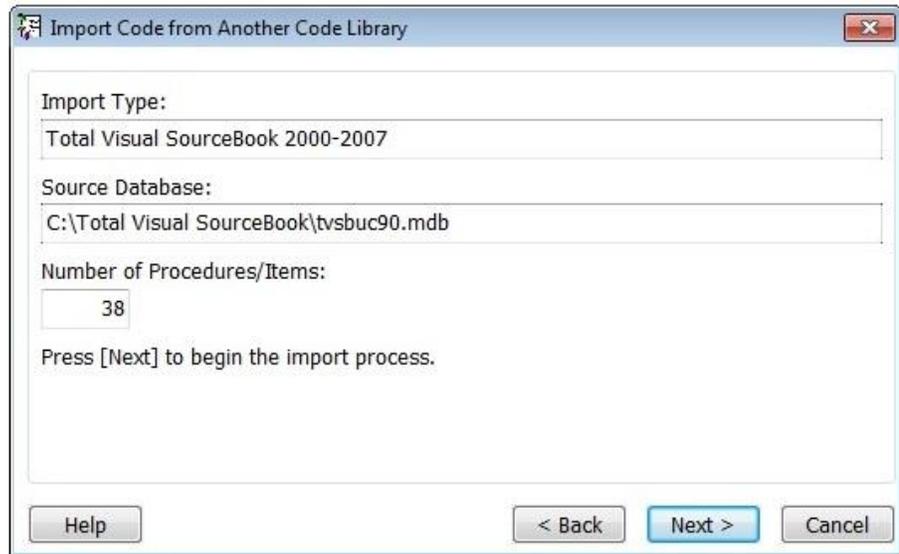


Select Location

If the previous version of the SourceBook product is installed, Total Visual SourceBook automatically detects the file location. Otherwise, use the [Browse] button to find the file location manually.

After specifying the location, click [Next] to continue.

The next form displays details about the file you selected, including Import type, location, and number of items to import:



Confirm Selection

If the file is invalid or cannot be opened, an error message appears on this form. To resolve this, press [Back] and point to a valid file. If there is no error message, click Next to finalize the import.

Once the code is imported, it is shown in the Total Visual SourceBook treeview under its own Topic and Category. If you wish to move the code to another location you can use “Save As” to save the code into a new location.

Chapter 8: Product Support

This chapter provides information on troubleshooting problems that arise and obtaining support for Total Visual SourceBook.

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 Visual SourceBook 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), forums, 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:

<http://support.fmsinc.com>

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.

Frequently Asked Questions (FAQs)

Common questions and additional information beyond what is in the manual is often available from our FAQs.

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 forums. 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 Forums	✓	✓	✓
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	Fee Per Incident	Included
<ol style="list-style-type: none"> 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 15.0 to 15.1) and are different from new builds that correct problems in existing features (e.g. from version 15.00.0001 to 15.00.0004).

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 2010 to Access 2013) 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 Visual SourceBook.

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 forums, 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 Visual SourceBook at:

<http://www.fmsinc.com/register>

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 Total Visual SourceBook. FMS does not, however, provide technical support or customer service for users of your applications that use Total Visual SourceBook code— this is your responsibility.

Contact Us

The best way to contact us is to submit a ticket on our support site:

<http://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.

Our ticketing system will let you track the progress of your issue and see the entire thread of communications and file attachments.

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

#

#M#, 56
#P#, 56
#X#, 57

A

Add-In, 32
adding items, 72
 new category, 74
 new class, 75
 new member, 77
 new module, 75
 new procedure, 77
 new topic, 72

B

backups, 68
bookmarks, 38, 43, 50–52
 creating, 51
 deleting, 52
 editing, 52
by reference, 27
by value, 27

C

class, 27
class dependencies, 21
code conflicts, 22
code database
 options, 55
code dependencies, 20–21

Code Explorer, 31–63
 bookmarks tab, 43
 code tab, 41
 details tab, 42
 example tab, 42
 export, 44
 information window, 41
 notes tab, 42
 opening, 32
 search results tab, 43
 toolbar, 35
 tracking window, 43
 treeview, 38
code group, 33
 join, 34
Code Librarian, 89
Code Repository, 5, 67–84
 adding items, 72
 delete code, 84
 editing, 84
 import code, 80
 making backups, 68
 overview, 68
commenting, 27
compact, 65
conflicts, 22
consistency, 20
conventions, 22–24
 controls, 22–24
 scope, 22–24
copy code, 37, 44
created by, 71
creation date, 71

D

delete code, 84

demos, 10
dependencies, 20, 21, 44, 45, 72
disabling error handling, 57
distribution rights, i, ii

E

edit, 84
editing conflicts, 69
error handling, 24–26, 56–60
 tokens, 56
error handling, 5
error handling module, 60
error handling options, 7
error handling selection, 59
examples, 28
Explorer. *see* Code Explorer
export code, 37, 44
 clipboard, 46
 file, 46
 notepad, 46
 project, 46
export options, 60

F

FMS Code Database, 55
FMS web site, 10–11, 94

G

gcfHandleErrors, 57

H

highlights, 4

I

import, 80
 SCE file, 87
information window, 41
installation, 13–18
 requirements, 14
 upgrading, 14
Internet web site, 10–11

J

join code group, 34

L

licensing, i–iii

M

menu, 35
Microsoft Office Developer Code Librarian, 89
modification date, 72
modified by, 72
module, 27
multi-user, 68, 69

N

naming conventions, 22–24
 controls, 22–24
 scope, 22–24
no error handling module, 60
notes
 updating, 62

O

options, 53–63
 code database, 55
 error handling, 56–60
 export, 60
 fonts, 62
 syntax highlighting, 63
 tab width, 61
 treeview, 63
 window size, 62

P

permissions, 9
programming standards, 20, 26
 consistency, 20
 naming conventions, 22–24
publishing library code, ii

R

read only, 63, 70, 77
readme file, 94
redistribution rights, i
refresh, 36, 70
registration, 11, 97
repair, 65

replace user name, 65
Repository. *see* Code Repository
Rich Text Editor, 73, 75, 76, 79

S

save as, 37, 61, 82, 83
search, 6, 36, 47–50

- boolean operators, 48
- category, 49
- filters, 49
- options, 49
- results, 50
- results tab, 43
- search phrase, 48

select default error handling, 7, 59
Source Code Exchange (SCE), 85–92

- creating SCE files, 86
- importing SCE files, 87

SQL Server database, 9
standalone, 32
standards, 20, 26

- consistency, 20
- naming conventions, 22–24

starting Total Visual SourceBook, 32
status bar, 35
storing code, 68
support, ii, 93
syntax highlighting, 63
system requirements, 14

T

tab width, 61
technical papers, 11
technical support, ii, 93, 97

Microsoft, 98
tokens, 56
toolbar, 35

- icons, 35–38

Total Access Memo, 73, 75, 76, 79
Total Access SourceBook, 10, 14
Total VB SourceBook, 10, 14
Total Visual SourceBook, 14
Total Visual SourceBook 2000, 10
tracking window, 43
treeview, 38
troubleshooting, 94
TVSBIC.MDB, 17, 55
TVSBUC.MDB, 17, 55

U

un-installing, 17
updates, 11
upgrading, 14
upsizing, 9
user access control (UAC), 9
user code database

- SQL Server, 9

User Code Database, 55
user name replace, 65

V

Visual Basic project, 46

W

web site, 10–11, 94
Windows API, 24