

---

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

**Total Visual**  
**SourceBook**™

For Microsoft® Office  
and Visual Basic 6.0



[www.fmsinc.com](http://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 ©1996-2009 by Financial Modeling Specialists, Inc.*

*All Rights Reserved.*

*Microsoft, Access, Word, Excel, PowerPoint, Outlook, Windows, Windows for Workgroups, Windows NT, Visual Basic, and Visual Basic for Applications are registered trademarks of Microsoft Corporation.*

*All other trademarks are trademarks of their respective owners*

*Printed in the USA.*



## *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 and John Litchfield
- **Documentation:** Luke Chung, Scott Ellis, Dan Haught, and Molly Pell

# Table of Contents

<b>Chapter 1: Introduction</b> .....	<b>3</b>
About Total Visual SourceBook .....	4
Product Highlights .....	4
New Features in This Version .....	6
Visit Our Web Site.....	8
<b>Chapter 2: Installation and Startup</b> .....	<b>11</b>
System Requirements .....	12
Installation Overview .....	12
Upgrading from a Previous Version .....	12
Program and Code Groups .....	13
Installation Strategies .....	15
Installing Total Visual SourceBook.....	17
Using the Update Wizard.....	17
Uninstalling Total Visual SourceBook .....	18
<b>Chapter 3: Code Style, Conventions, and Usage</b> .....	<b>21</b>
Coding Overview.....	22
Consistency.....	22
Dependencies .....	22
Avoiding Conflicts with Your Code.....	24
Naming Conventions.....	24
Error Handling.....	26
Structured Programming Style and Formatting.....	28
Using Code Updates.....	28
Other Issues .....	29
Example Code .....	30
<b>Chapter 4: Code Explorer</b> .....	<b>33</b>
Running the Code Explorer .....	34
Using the Code Explorer .....	37
Toolbar Icons .....	38
Code Explorer Windows .....	41
Getting Code into Your Project.....	46
Printing Code .....	49
Finding Code with the Search Tool .....	50
Using the Search Results.....	52
Using Bookmarks .....	53
Setting Options .....	54
Administrative Features.....	62

<b>Chapter 5: Code Repository</b> .....	<b>65</b>
Code Repository Overview .....	66
Importance of Backups.....	66
Multi-User Issues .....	67
System-Maintained Information .....	69
Adding New Items .....	70
Importing Code.....	78
Saving Classes, Modules, and Procedures .....	80
Editing Items.....	82
Deleting Code .....	83
<b>Chapter 6: Source Code Exchange</b> .....	<b>85</b>
About Source Code Exchange.....	86
Creating an SCE Package.....	86
Importing an SCE Package .....	88
Importing Data from another Code Library.....	89
<b>Chapter 7: Product Support</b> .....	<b>93</b>
Troubleshooting .....	94
Web Site Support.....	94
Technical Support Options .....	95
Contacting Technical Support.....	97
<b>Index</b> .....	<b>99</b>



## ***Welcome to Total Visual SourceBook***

Thank you for selecting Total Visual SourceBook, the most popular professional source code library for VB/VBA developers. Total Visual SourceBook supports VB/VBA developers regardless of platform, so whether you are in Access, VB, 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. In addition to Total Visual SourceBook, we offer a wide range of products for Microsoft Access and VB 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)

We also offer many award winning products for SQL Server and Visual Studio .NET. Visit our web site, [www.fmsinc.com](http://www.fmsinc.com), for more information.

If you didn't purchase Total Visual SourceBook directly from us, please be sure to register on line and sign up for our free email newsletter. This guarantees that you are contacted in the event of upgrades, news, 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**
- **New Features in This Version**
- **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 over 45 categories, 180 classes and modules, 2800 procedures, and 95,000 lines of code, every major area of application development is covered:

- Forms and Controls
- Error Handling
- Printing
- Access 97-2007
- XML
- Excel 97-2007
- Internet and Web
- Math Routines
- Common Dialogs
- Access, Word and Excel
- Soundex and Metaphone
- Compression Algorithms
- Parse and Tokenize Text
- Visual SourceSafe
- Clipboard Operations
- INI and Help Files
- Abstract Data Types
- Database: ADO, RDO, Jet, ODBC and DAO
- File and Disk Operations
- Graphics
- Word 97-2007
- Outlook 97-2007
- Word AutoConvert
- Statistics and Probability
- Fonts
- Multimedia Extensions
- Searching and Sorting
- Password Encryption
- String Handling
- Date and Time
- Data Type and Unit Conversions
- Application Techniques
- Windows Registry
- Rotated Text and Bitmaps

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 Visual Basic and VBA**

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 2007. 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 Options** on page 59 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.

---

## **New Features 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 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 VB and VBA developers.

### **What's New in This Version**

This version of Total Visual SourceBook has been enhanced to support 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 multivalued 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](http://www.fmsinc.com) to find out about new products and updates.

### Product Registration

If you did not order Total Visual SourceBook directly from us, you can register it online. 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, or see **Using the Update Wizard** on page 17 for information on how to check for the availability of 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.

### Technical Papers and Tips

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.

### Blog and Newsgroups

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

[www.fmsinc.com/blog](http://www.fmsinc.com/blog)

Visit our newsgroups at:

[www.fmsinc.com/support/newsgrp.htm](http://www.fmsinc.com/support/newsgrp.htm)

Or see our web site for additional instructions.

### **Links to Other Development Sites**

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



# 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, and instructions for uninstalling.*

---

## Topics in this Chapter

- **System Requirements**
- **Installation Overview**
- **Upgrading from a Previous Version**
- **Program and Code Groups**
- **Installation Strategies**
- **Installing Total Visual SourceBook**
- **Using the Update Wizard**
- **Uninstalling Total Visual SourceBook**

---

## System Requirements

Total Visual SourceBook is written in Visual Basic, and its requirements are similar to those of Visual Basic itself:

- Hardware and operating system supported by Microsoft Access and Visual Basic.
- 15 MB free disk space to install the product, and additional disk space to store your code databases.
- Internet Explorer 4.01 or higher.

---

## Installation Overview

Total Visual SourceBook offers a great deal of flexibility in how it stores code and makes code available to multiple developers. Because the Code Repository feature is multi-user ready, you can install and configure the product to allow access to multiple developers. Alternatively, you can install the program on a single user computer so only you can access the code.

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.

Don't be too worried about making an incorrect choice when you install the product—Total Visual SourceBook is designed to be flexible, and you can make changes later or re-run the setup to change your configuration.

---

## 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. Install the new version to a new folder (do not overwrite your existing version), and use the Code Group Options to point to your existing user code. See page 58 for details about connecting to a new user code database.

### Upgrading from Total Access SourceBook/Total VB SourceBook

If use Total Access SourceBook or Total VB SourceBook, you can import your user code into Total Visual SourceBook.



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.

---

## Program and Code Groups

Total Visual SourceBook consists of two main components:

- **Code Explorer**  
The program that you run to view, search, and edit code.
- **Code Group**  
The set of two databases that store the FMS source code and the code you add to the Repository.

The Code Explorer consists of a Visual Basic executable, various DLL and OCX files, and entries in the system registry. Like Visual Basic itself, the program must be installed on a local computer for it to run. You cannot install it on a network drive and have other users to run it directly from there.

The two databases that comprise the Code Group can be installed on a local or shared drive. The FMS and User code databases can reside in different directories.

### How the Code Group Works

Total Visual SourceBook stores source code in a Code Group that consists of two Microsoft Access (Jet 3.5 format) databases:

- The built-in code (the source code that ships with the product) is stored in the FMS Code database. This file is TVSBIC90.MDB.
- The code you add through the Code Repository is stored in the User Code database. This file is TVSBUC90.MDB.

One of the most powerful features of Total Visual SourceBook is that you can create and use multiple Code Groups. For example, you can install the product on your laptop computer, and on a shared network drive. In this configuration, you have access to the built-in code, along with your own code even when you are not connected to the network. You can also

connect to a Code Group stored on the network to access your organization's shared code assets.

In another configuration, you can install the product on your local computer and create several Code Groups on your local drive to manage code for each of your projects. Of course, this scenario eliminates the usefulness of a common shared repository of code, but does allow compartmentalization if you need it.

To manage the use of multiple Code Databases, Total Visual SourceBook offers a Join option. This option allows you to "join" any Code database that you can see on your file system. For more details on using the Join option, see **Code Group Options** on page 58.

### Where to Install the Code Databases

When you run the Setup program, you are prompted for the location of the Code Group. Remember that the Code Group consists of the FMS Code database and the User Code database that stores your code. The answer to this question determines whether or not you want other developers using Total Visual SourceBook to be able to access the code in a shared manner.

- If you install the Code Group in a non-shared directory, such as a hard disk on your local machine, you are creating a single-user installation. In such a case, you are the only person who can access the Code Group.
- If you install the Code Group in a shared directory, such as a network drive, other Total Visual SourceBook users can access the built-in and user code. Such an installation is ideal for multi-developer teams.



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

### Multi-User Database Access

Each developer using Total Visual SourceBook must have a licensed copy installed. When you attempt to access a shared Code Group, the program checks to ensure that no other developers are using your licensed copy. If the program detects a conflict, it disallows access to the shared Code Group. For example, if two developers install Total Visual SourceBook using the same license, those two developers cannot access the same shared Code Group at the same time.

Because you may have two or more copies of your VBA host open at the same time, Total Visual SourceBook allows multiple copies of the program to run on the same computer. Of course, once you start loading multiple copies of large programs like Visual Basic, you can run out of resources or experience severe performance degradation. Total Visual SourceBook, however, places no restrictions of its own on the number of concurrent instances per machine.

Please refer to the printed license agreement at the beginning of this User Guide for details about licensing.

---

## Installation Strategies

This section outlines common installation scenarios and helps you decide how to install the product.

### Simple Single User Installation

This is the simplest configuration. Install the program and the Code Group on your local drive. In this configuration, all files are local and the Code Group is not shared. This is also an ideal configuration for use when you are evaluating the functionality of Total Visual SourceBook for possible future use in a multi-user environment.

### Simple Multi-user Installation

This installation allows multiple developers to share a single Code Group. In this scenario, you install the program and specify a shared drive for the Code Group. Setup then installs the program on your local computer, and installs the Code Group on the shared drive. You are now able to run the product on your computer, and the built-in code and user code is stored on the shared drive.

Later, each additional developer who wants to use Total Visual SourceBook runs the setup program with a unique license (serial number). When prompted for the Code Group location, that developer points to the shared network drive. The Setup program detects an existing Code Group and does not overwrite it. The Total Visual SourceBook program files are copied to the developer's machine. Repeat this step for each developer who wants to join the shared Code Group. Remember that the program requires each user to have a unique license (serial number) to access a shared Code Group.

## Coexisting Single User and Multi-user Installations

In this scenario, you perform a standard single user setup to install the program and Code Group to your computer. This allows you to keep the user code on your machine private. Later, you can use the Join option available in the Code Explorer to point your installation of Total Visual SourceBook to different a Code Group stored on a shared drive. This way, you can switch back and forth between multiple Code Groups. See **Code Group Options** on page 58 for more information on the Join option. This scenario is also ideal if you are running the program from a laptop computer that may sometimes be disconnected from your networks. In this case, when Total Visual SourceBook starts up and detects that a network Code Group is not available, it prompts you to point to another Code Group. In this case, you would point the program to your local Code Group. Note that if you make changes on your local Code Group, they are not automatically replicated to your network Code Group when you reconnect. In such a case, use the Source Code Exchange feature to replicate your data changes. See **Chapter 6: Source Code Exchange** for more information about this feature.

## Creating a New Code Group

There may be situations when you want to install a Code Group on a specific drive, but already have Total Visual SourceBook installed on your computer. To do this, re-run the Setup program again and specify the new directory for the Code Group. This re-installs the program on your computer, and creates the new Code Group in the directory you specify. Note that this does not change any Code files located on your local drive—it merely creates new Code Group databases.

## Single User Installation on a Shared Drive

Even if you are not working in a multi-user environment, you may want to consider installing your Code Group to a shared drive that is part of a backup/archive regimen maintained by your organization.

For most developers, of course, local disk access is almost always faster than network disk access, so the performance penalty you pay while using Total Visual SourceBook may not be worth the convenience of a shared backup mechanism. If network performance is poor, install the Code Group locally and institute your own backup process. For more information on making backups, see page 66.

---

## Installing Total Visual SourceBook

This section contains the information you need to install Total Visual SourceBook on your computer.

### File Updates

Total Visual SourceBook is compiled using Visual Basic 6.0, Service Pack 3. This means the setup program may install updated versions of the VB runtime module (MSVBM60.DLL), the Common Control OCX (COMCTL32.OCX), and several other files on your machine.

Before overwriting an existing shared component on your system, the Setup program verifies that its version is a later version than the one you have. Additionally, when you run the Setup program, you have the option to make backups of any replaced files. We recommend that you select this option so that you can restore your system to its prior state if necessary. For more information on uninstalling the program, see page 17.

### Installation

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

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

### List of Files Installed

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

---

## Using the Update Wizard

Total Visual SourceBook includes a built-in mechanism to check the availability of updates via the Internet. If you have an active Internet connection, you can use the Total Visual SourceBook Update Wizard to ensure that you have the latest version.

To run this program, press the Windows Start button and select Programs, FMS, Total Visual SourceBook, Update Wizard. Follow the prompts to check for the latest update.

Total Visual SourceBook supports two types of updates:

- **Program Service Pack**  
This type of update contains fixes and enhancements to the program itself: the Code Explorer, Code Repository, and on-line help.
- **Code Service Pack**  
This type of update contains enhancements to the library code and notes included with the product.

---

## 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.

The Total Visual SourceBook uninstall process does not delete the code databases. 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 follows standard Windows installation protocol, so uninstalling is straightforward:

From the Windows Start Menu, select Control Panel, then

#### *Windows Vista*

1. Select Uninstall a Program (in Programs section)
2. Select FMS Total Visual SourceBook from the list of installed programs
3. Click on Uninstall/Change

### **Windows XP**

1. Select Add/Remove Programs
2. Select FMS Total Visual SourceBook from the list
3. Press the Change/Remove button
4. Then, follow the prompts to uninstall.

After a few moments, the Total Visual SourceBook program files and its registry entries are deleted. The program no longer appears in either your Start menu, or the VB/VBA Add-Ins menu.

Please note that uninstalling does not restore backups of shared system components such as OCX and DLL files. Any system components that were replaced during installation are stored in the backup folder you specified. If you want to restore your system to its original state, you can move the files from the backup folder back to their original location. Before doing so, however, you should verify that no other programs on your system are relying on the new versions installed by Total Visual SourceBook.



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**
- **Using Code Updates**
- **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 45), 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 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.

The code available in Total Visual SourceBook uses the following variable naming conventions.

Variable Type	Tag	Example
Boolean	f	fIsLoaded
Byte	byt	bytOffset
Currency	cur	curAmountDue
Date	dat	datExpires
Double	dbl	dblOffset
Enumerated Type definition	Enum	EnumJetObjectTypes

Variable Type	Tag	Example
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 Options** on page 59 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 procedure 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 GenericFunction() as Boolean
' Comments : Comments about GenericFunction
' Parameters: None
' Returns : True or False
' Source : Total Visual SourceBook
'
On Error GoTo PROC_ERR

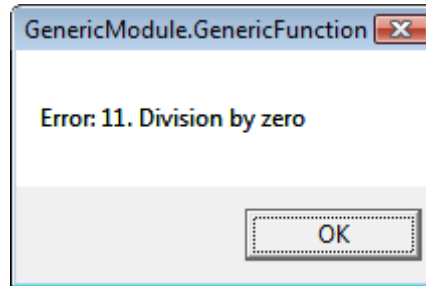
Debug.print 1 / 0

PROC_EXIT:
Exit Function

PROC_ERR:
MsgBox "Error: " & Err.Number & ". " & Err.Description, , _
"GenericModule.GenericFunction"
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 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 Options on page 59 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.

---

## Using Code Updates

One of the main benefits of Total Visual SourceBook is that you can use the source code as is, or customize it for your specific needs. This does, however, have one drawback.

FMS may provide occasional Code Service Packs with code enhancements and fixes. If we release a newer version of code that you have used, you may need to update your projects.

If you used the FMS code without modifications, this is as simple as overwriting the modified procedure. However, if you customized the code, you don't want to overwrite your version or you'll lose your changes.

When you customize FMS source code, you should take the time to document your changes. Later, if there is a newer version available in a Code Service Pack, you can easily determine which code is in its original state from Total Visual SourceBook, and which is modified.

---

## 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.

Additionally, if the procedure changed, the comment block contains information on the update(s). As we update procedures, we make the changes available to registered users in Code Updates available from our web site.

```
' Source      : Total Visual SourceBook  
' Updated    : Code Service Pack 2
```

If you apply a Code Update to your Total Visual SourceBook installation, you want to be sure your applications contain the latest version of Total Visual SourceBook code. Rather than doing a line-by-line comparison of the old and new versions, you can simply check your project's comment code against the updated version's comments.

### 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 VB/VBA itself, or to replace standard VB/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 45 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 VB 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 47).
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 'cmdFlat'
' 4. Paste all the code from this example to the new form's module.

Private Sub cmdFlat_Click()
    Static fFlat As Boolean

    fFlat = Not fFlat

    ' Example code for SetToolBarFlat
    ' This code toggles the toolbar style between regular mode
    ' and 'flat' mode

    SetToolBarFlat Toolbar1, fFlat
End Sub

Private Sub Form_Load()
    Dim tbn As ComctlLib.Button

    With Toolbar1
        Set tbn = .Buttons.Add(, , "Button 1")
        Set tbn = .Buttons.Add(, , "Button 2")
        Set tbn = .Buttons.Add(, , "Button 3")
        Set tbn = .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**
- **Printing Code**
- **Finding Code with the Search Tool**
- **Using the Search Results**
- **Using Bookmarks**
- **Setting Options**
- **Administrative Features**

---

## 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.

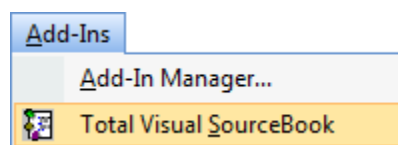
Alternatively, you can execute the file named TVSB.EXE located in the program folder you specified during installation.

#### ***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.



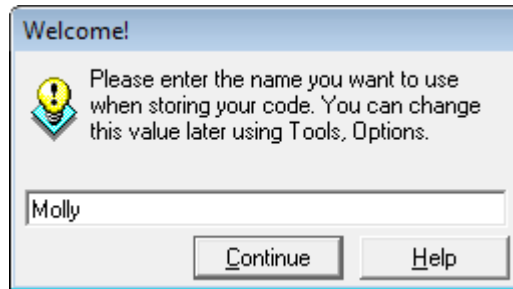
*VBA IDE Add-Ins menu*

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.

## User Login Form

The first time you run Total Visual SourceBook, the User Login form appears:



*Total Visual SourceBook Login Form*

Use this form to identify yourself to the Total Visual SourceBook program. Your user name is very important because Total Visual SourceBook uses it to identify you as the owner of the procedures you add using the Code Repository.

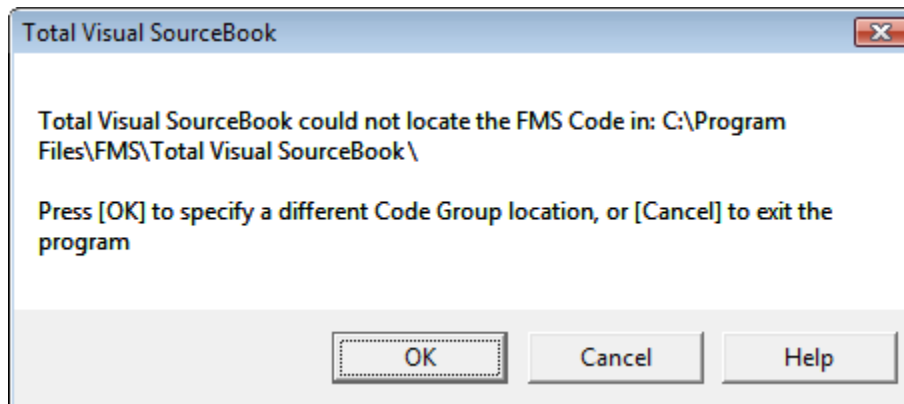
The Login form initially shows your Windows user name. You may type in any name you wish to use as your identifier to the program. It is generally a good idea to use the same user name that your installation of Windows uses, or your network login name. This eliminates the confusion of having multiple user names. Once you enter your user name, press the [Continue] button and the Code Explorer program continues to load.

The next time you start the program, the Login form is bypassed, and the user name you initially specified is automatically used. You can change your User Name at any time using User Name Replace feature, explained on page 62.

## 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*

Press the [OK] button 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 62) 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.

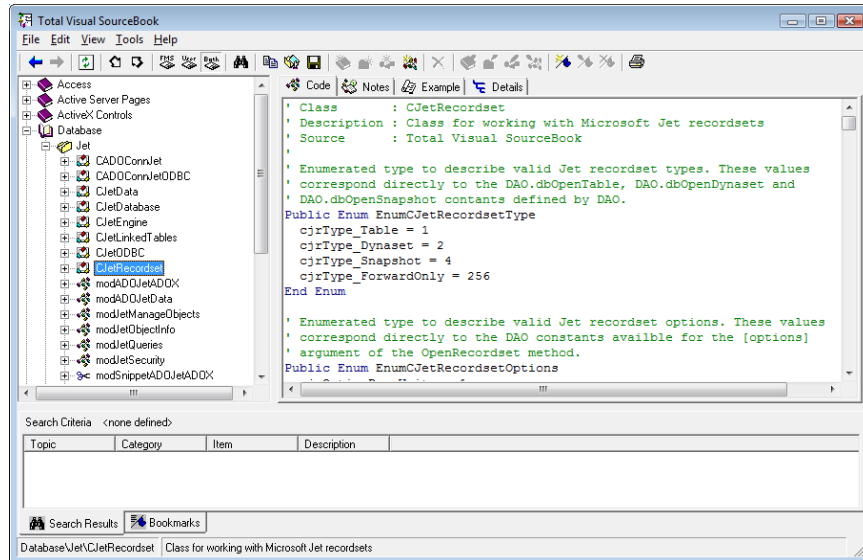
### **Joining Other Code Groups**

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 58. For more information on Code Groups and how to use them effectively, see page 13.

---

## 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.
- The **Information Window** shows detailed information about the selected item. This includes code, notes, and examples, and details.
- The **Tracking Window** at the bottom shows Search Results and all the Bookmarks you defined.

### Toolbar and Menus

Total Visual SourceBook offers many features that make it easy to find, manage and export code. These features are accessible through three areas:

- The program menu at the top of the window.
- The program toolbar just below the menu. You can customize the toolbar by selecting View, Toolbars, Customize from the menu.
- Context menus available by right-clicking on various objects in the window.

## Status Bar

The status bar at the bottom of the Code Explorer displays the path and description for the current item:

Database\Jet\CJetRecordset | Class for working with Microsoft Jet recordsets

*Status Bar*

---

## Toolbar Icons

The Toolbar appears just below the menu:



*Total Visual SourceBook Toolbar*

The following functions are available from the Toolbar:

### Display Icons

The first set of icons on the Toolbar are Display Icons.

#### **Back**

Also available from the View, Back menu, this feature moves to the previously selected item in the Treeview. A history list is maintained so you can move back several levels.

#### **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.

#### **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 in a shared Code Group 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.

#### **Expand All**

Also available from the View, Expand All menu, this feature displays all items in all categories.

### **Collapse All**

Also available from the View, Collapse All menu, this feature collapses the Treeview items so only categories 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 shows all the FMS and user code.

## **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 50.

### **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 46.

### **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 47.

## **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 **Chapter 5: Code Repository**, page 65.

### **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.

### **Delete**

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

### **Edit Topic**

Also available from the Edit, Topic menu, this feature opens the Total Visual SourceBook Edit Topic form to edit the topic currently displayed in the Treeview. You may only edit user-defined topics, not FMS topics.

### **Edit Category**

Also available from the Edit, Category menu, this feature opens the Total Visual SourceBook Edit Category form to edit the category currently displayed in the Treeview. You may only edit user-defined categories, not FMS categories.

### **Edit Class or Module**

Also available from the Edit, Class/Module menu, this feature opens the Total Visual SourceBook Edit Class/Module form to edit the current class or module displayed in the Treeview. You may only edit user-defined code, not FMS code.

### **Edit Member or Procedure**

Also available from the Edit, Member/Procedure menu, this feature opens the Total Visual SourceBook Edit Category form to edit the current class member or procedure displayed in the Treeview. You may only edit user-defined code, not FMS code.

### **Bookmark Icons**

The next set of icons helps you create and manage bookmarks. For more information on bookmarks, see **Using Bookmarks** on page 53.

### **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.

### **Print**

### **Print**

Also available from the File, Print menu, this icon prints the currently displayed item in the Code Explorer. For more information, see **Printing Code** on page 49.

---

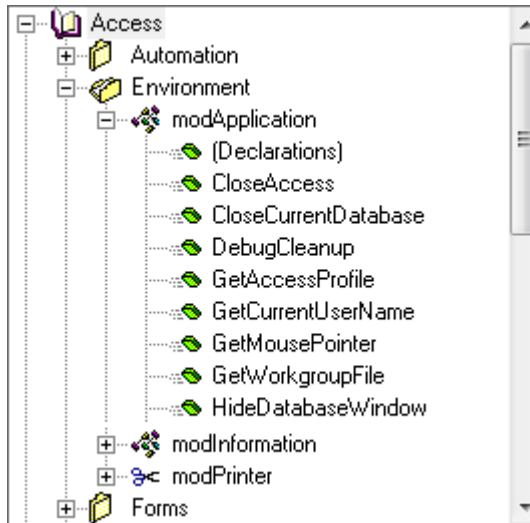
## Code Explorer Windows

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:

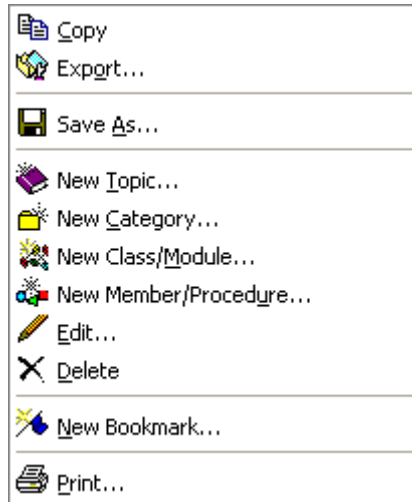


When you select a particular item in the Code Explorer Treeview, the code is displayed in the Code Window to the right. If you select a category rather than a code item in the Treeview, a short description of the category is displayed in the Code Window.

### ***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. See the toolbar section (page 38) for more information on each menu item.



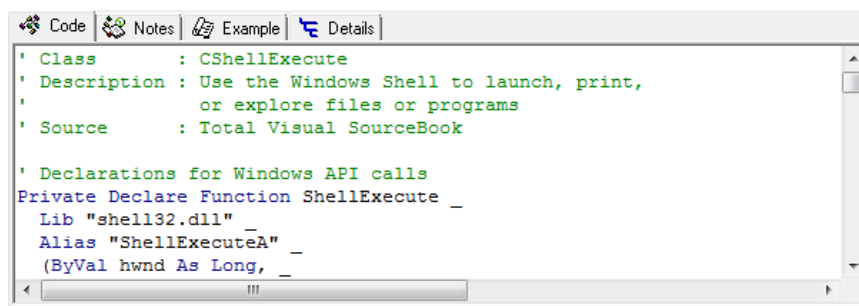
*Treeview Right-Click Popup Menu*

## 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:



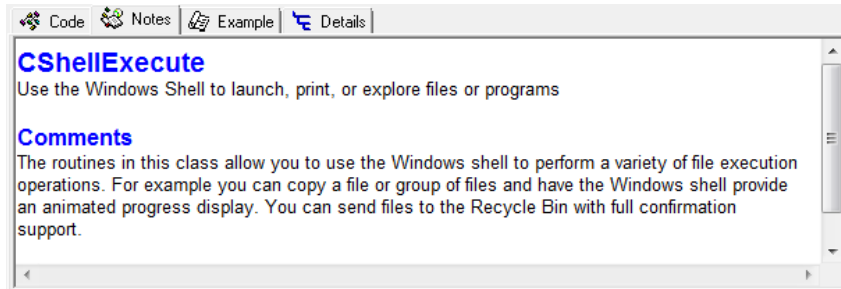
*Code Tab*

If the Code Explorer is currently on a topic or category rather than on a code item, the Code Tab is empty. This is because 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

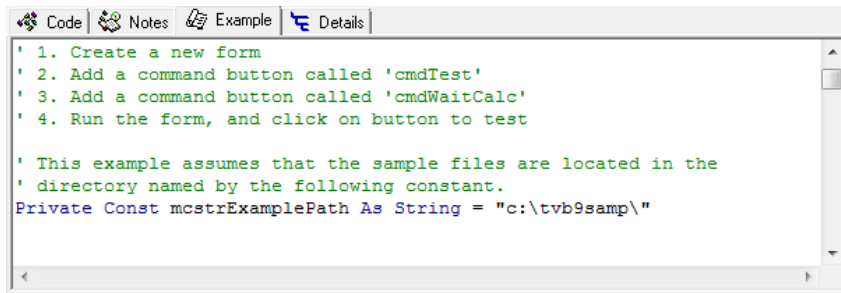
The Notes tab shows detailed information about the selected item:



Notes Tab

### Example Tab

The Example tab provides an example for using the current class or module:



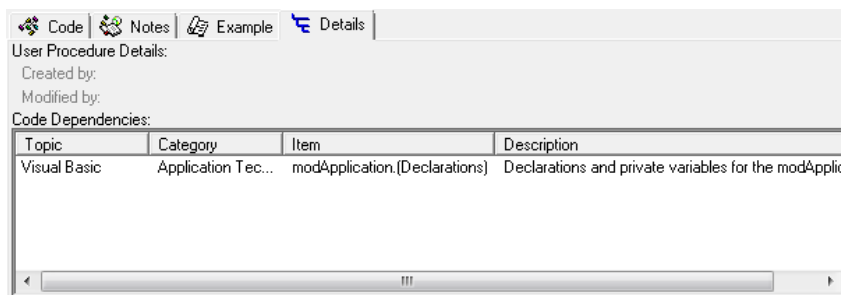
Example Tab

If you are on an individual procedure, the example for the entire class or module is displayed. This tab is empty if you are on a category.

For more information on using the Example code, see page 30.

### Details Tab

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



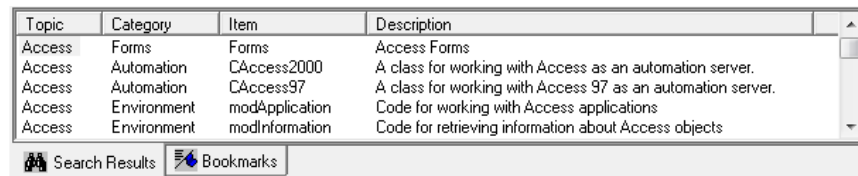
Details Tab

## 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 **Using the Search Results** on page 52.

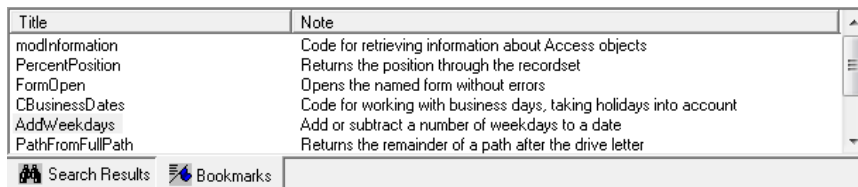


Topic	Category	Item	Description
Access	Forms	Forms	Access Forms
Access	Automation	CAccess2000	A class for working with Access as an automation server.
Access	Automation	CAccess97	A class for working with Access 97 as an automation server.
Access	Environment	modApplication	Code for working with Access applications
Access	Environment	modInformation	Code for retrieving information about Access objects

Search Results Tab

### Bookmarks Tab

The Bookmarks tab displays the bookmarks set on code. For more information, see **Using Bookmarks** on page 53.



Title	Note
modInformation	Code for retrieving information about Access objects
PercentPosition	Returns the position through the recordset
FormOpen	Opens the named form without errors
CBusinessDates	Code for working with business days, taking holidays into account
AddWeekdays	Add or subtract a number of weekdays to a date
PathFromFullPath	Returns the remainder of a path after the drive letter

Bookmarks Tab


---

## Getting Code into Your Project

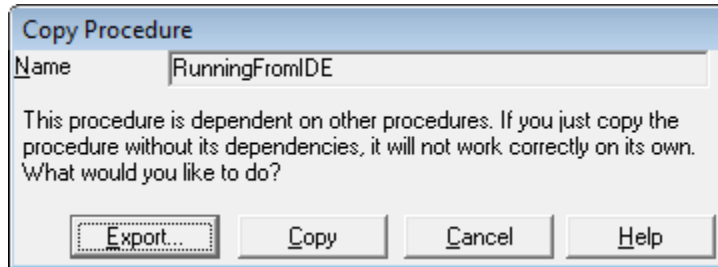
Once you find the class, module, or procedure you want, you need to get it into your project. There are two ways to do this:

- **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:




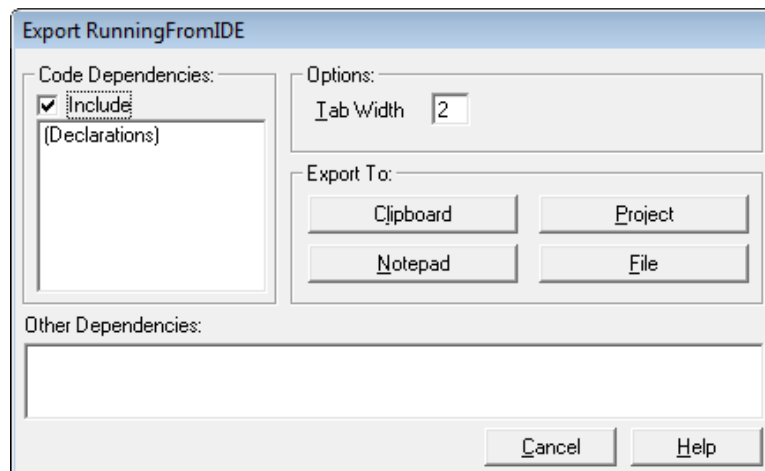
*Copy or Export Dialog*



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 was 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.

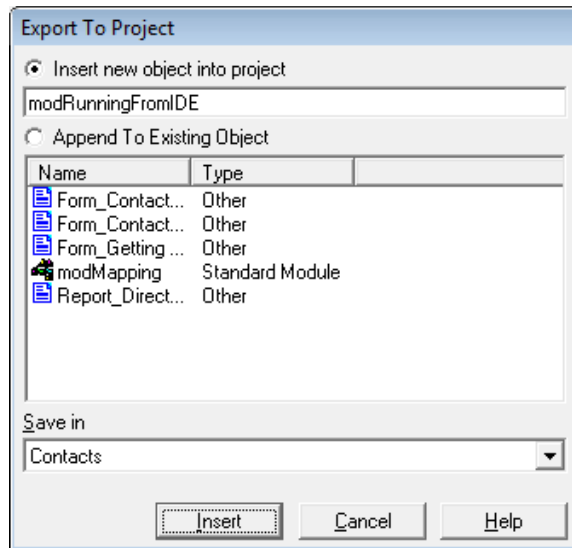
If you are using a VBA host (such as Access or Word), you can import the file by selecting File, Import File, and choosing the exported file.

### *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*

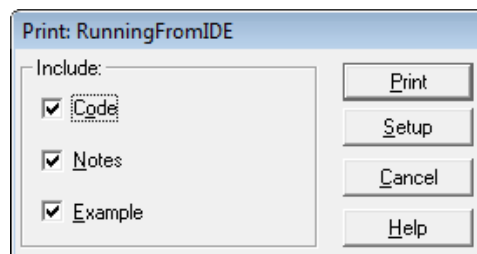
### **Save in**

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.

---

## **Printing Code**

To print the current procedure, press the  button on the toolbar or select File, Print from the menu. This feature displays the Print dialog.



*Print Dialog*

The Print dialog lets you specify any of three items to print:


- **Code**—Print the code tab displayed in the Code Window
- **Notes**—Print the notes tab displayed in the Code Window
- **Example**—Print the example tab displayed in the Code Window

Each of these items is printed on a separate page.

---

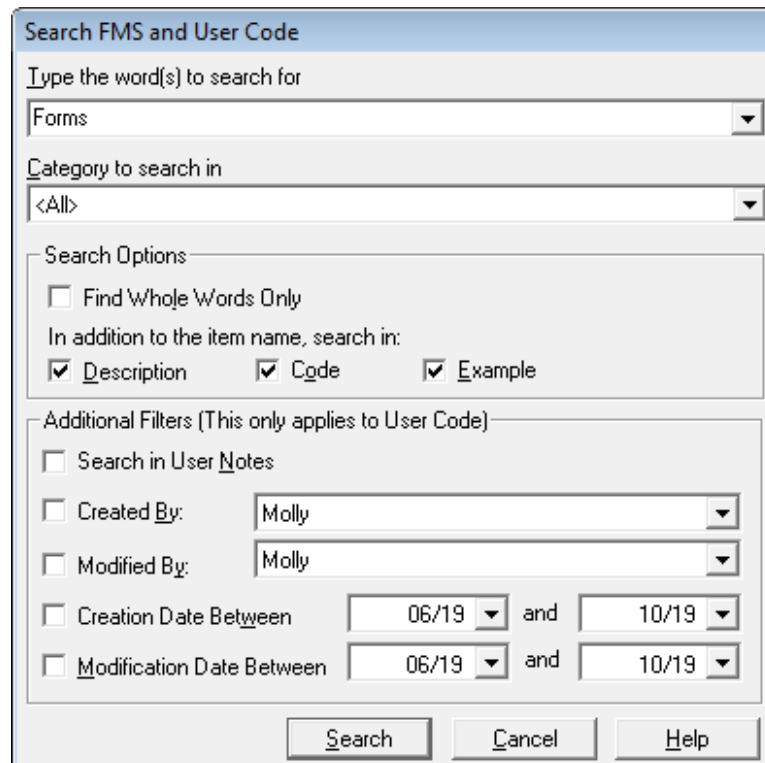
## Finding Code with the Search Tool

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

Forms

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: Molly

Modified By: Molly

Creation Date Between 06/19 and 10/19

Modification Date Between 06/19 and 10/19

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.

Topic	Category	Item	Description
Access	Forms	Forms	Access Forms
Access	Automation	CAccess2000	A class for working with Access as an automation server.
Access	Automation	CAccess97	A class for working with Access 97 as an automation server.
Access	Environment	modApplication	Code for working with Access applications
Access	Environment	modInformation	Code for retrieving information about Access objects

*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.

---

## 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.

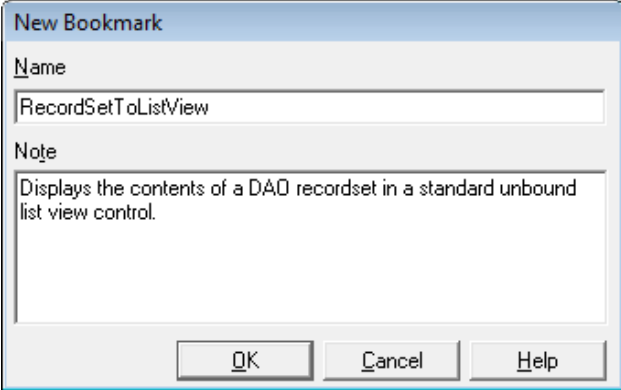
Title	Note
modInformation	Code for retrieving information about Access objects
PercentPosition	Returns the position through the recordset
FormOpen	Opens the named form without errors
CBusinessDates	Code for working with business days, taking holidays into account
AddWeekdays	Add or subtract a number of weekdays to a date
PathFromFullPath	Returns the remainder of a path after the drive letter

Search Results    Bookmarks

*Bookmarks Tab*

### Creating a Bookmark

To add a bookmark, select the 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:



The dialog box titled "New Bookmark" has two text input fields. The first is labeled "Name" and contains the text "RecordSetToListView". The second is labeled "Note" and contains the text "Displays the contents of a DAO recordset in a standard unbound list view control." At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

*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.



In the Note field, use [Ctrl+Enter] to add a line break.

### 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 62.

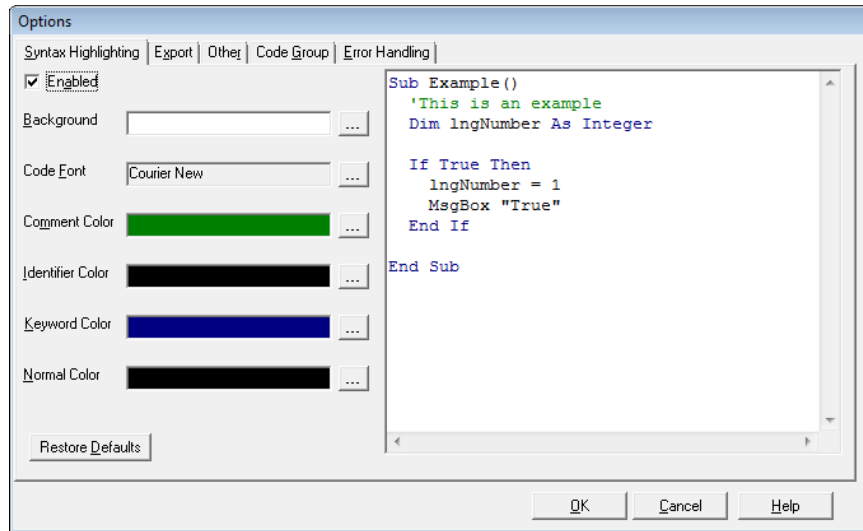
---

## **Setting Options**

You can specify a variety of settings to control the how the Code Explorer behaves, how exported functions are formatted, and what error handling code is used by default. To set options, select Tools, Options from the menu.

### **Syntax Highlighting Options**

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



Options Form, Syntax Highlighting Tab

### **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.

### **Background**

Specify the background color used to display code.

### **Code Font**

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

### **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 Visual Basic keywords.

### **Normal Color**

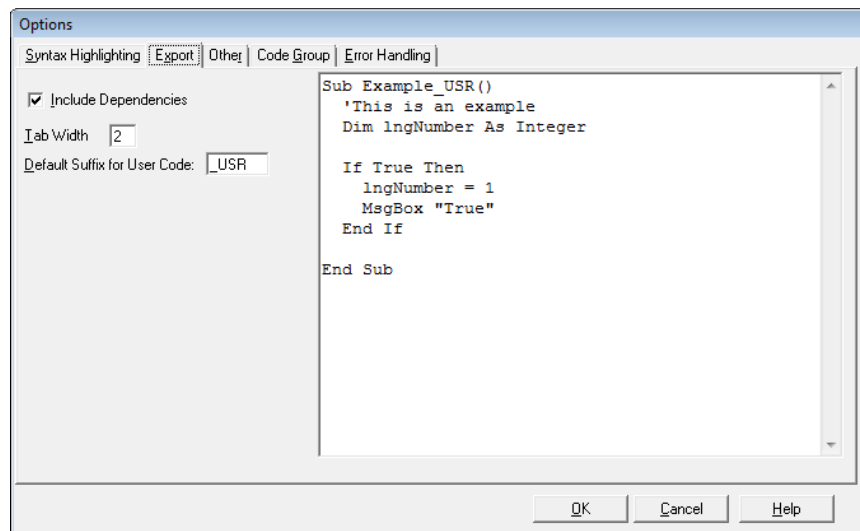
Specify the color for normal characters (words that are not identifiers, comments, or keywords). These include numbers, strings, and mathematical symbols.

## Restore Defaults

Restore the default values used by Total Visual SourceBook.

## Export Options

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 when Applicable***

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.

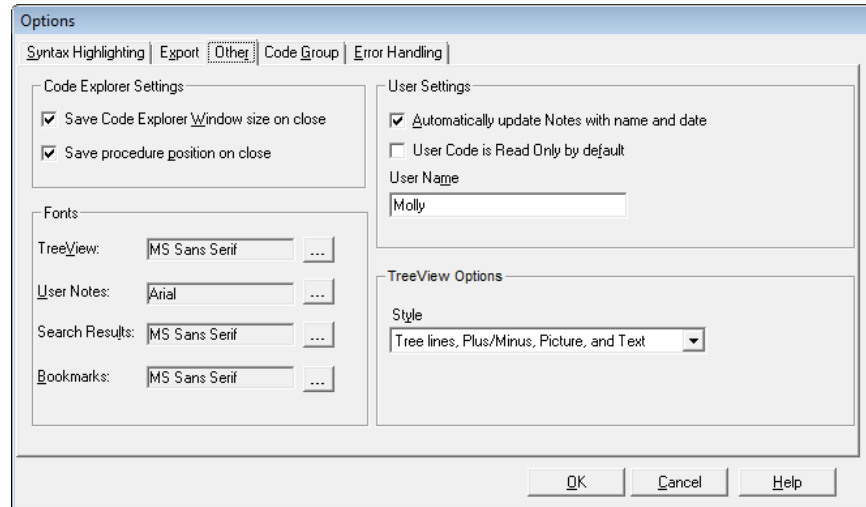
### ***Tab Width***

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.

## Other Options



Other 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 54).

### ***Automatically Update Notes with Name and Date***

Check this option 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** page 68.

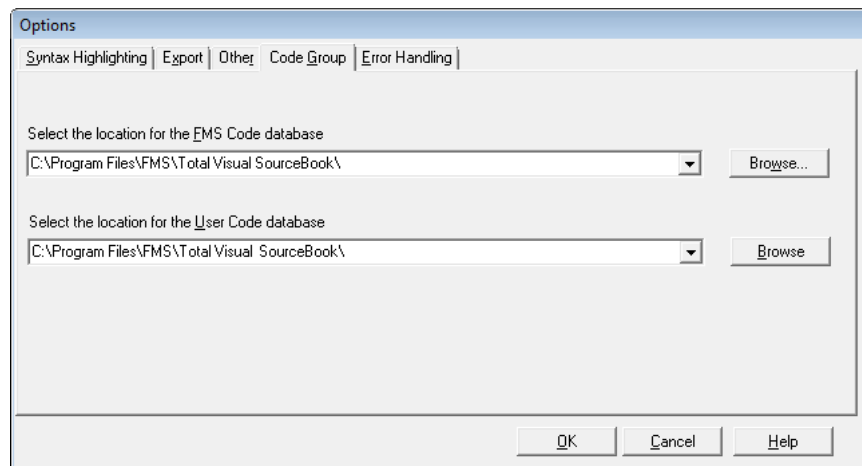
### **Treeview Options**

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

### **Code Group Options**

Total Visual SourceBook supports the ability to view different groups of code stored in different locations using Code Groups. A Code Group consists of the database containing FMS source code, and the database containing user created code. You can change the Code Group that Total Visual SourceBook uses on the Code Group tab of the options dialog:



*Group Tab of the Options Form*

For more information on creating and using Code Groups, see page 13.

There are two locations that you can specify:

- **FMS Code Database** – The location of the database containing the built-in FMS code ( TVSBIC90.MDB).
- **User Code Database** – The location of the database containing the code you or your team members have added ( TVSBUC90.MDB).

If you select a Code Group 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 group.



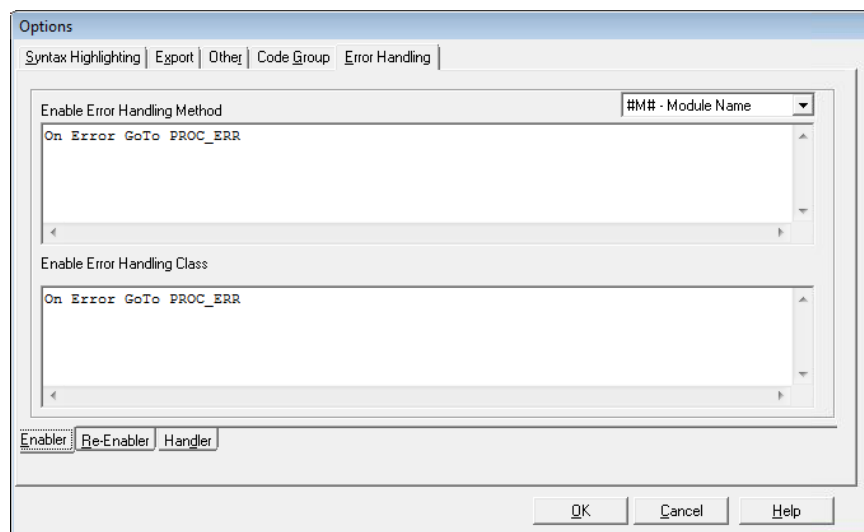
Note that you can only join a Code Group that was created by this version of Total Visual SourceBook. You cannot join a Code Group created by other FMS products, or earlier or later versions of FMS products.

## Error Handling Options

In general, every non-trivial subroutine, function, and property procedure 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 of the Options dialog lets you easily replace the default error handling code with your own. There are 3 sections for the error handler: Enabler, Re-Enabler, and Handler.

### Enabler

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



*Options Form, Error Handling Tab, Enabler*

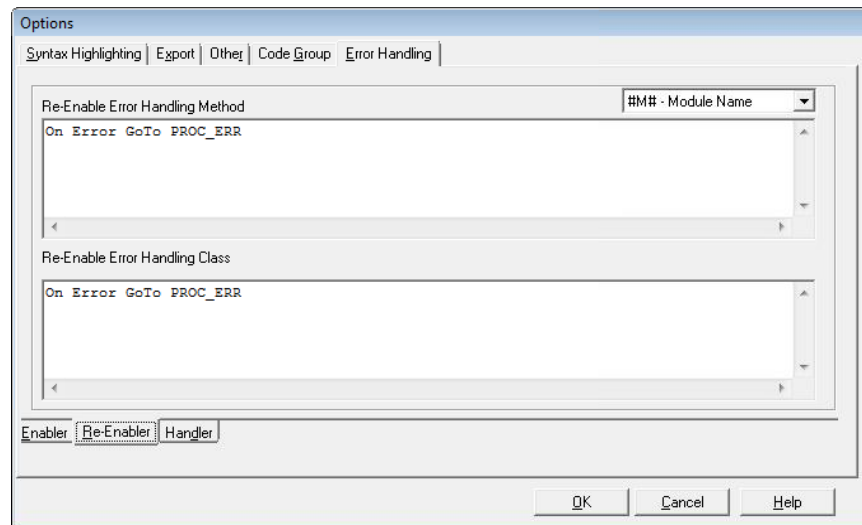
Use the On Error statement to instruct the application about what to do if it encounters an error. For example:

```
On Error GoTo PROC_ERR
```

This line is inserted in every procedure before the first error could potentially occur.

### **Re-Enabler**

The Re-Enabler tab includes the code to resume error handling after an On Error Resume Next statement:

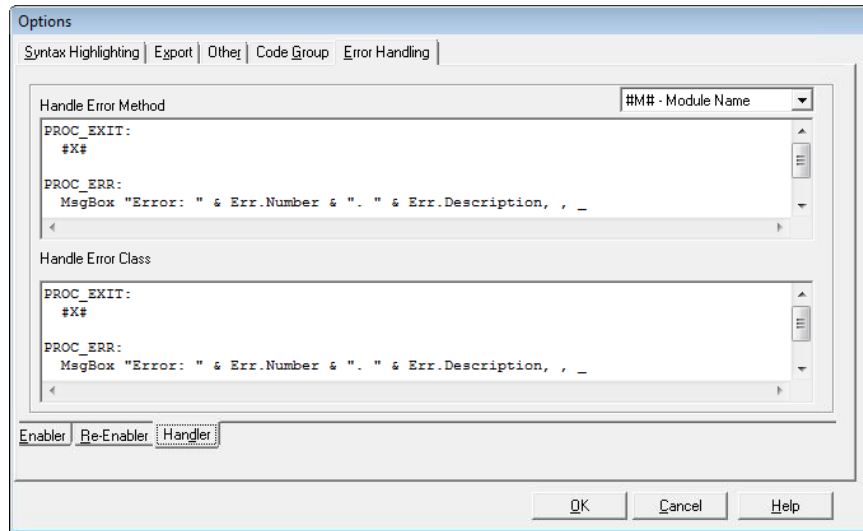


*Options Form, Error Handling Tab, Re-Enabler*

This is generally the same as the Enabler code, but may be different if desired.

### **Handler**

The Handler tab includes the error handling code:



Options Form, Error Handling Tab, Handler

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

### 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"
```

The following Tokens are available:

Token	Description
#M#	The module name.
#P#	The Procedure Name
#X#	Exit Sub or Exit Function (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—use the “Module” and “Class” tabs along the bottom to switch between the two.

---

## Administrative Features

Total Visual SourceBook offers several functions that make it easy to perform standard administrative functions.

### 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. The Repair operation is automatically performed if necessary.

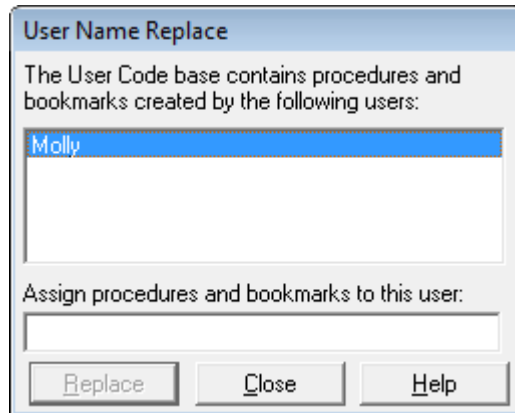


You should always perform a backup of your databases prior to compacting and/or repairing them. You can do this simply by copying the databases to another location, or by ensuring that the files are part of a daily backup routine. You may also want to consider our **Total Visual Agent** program that automates database repair, compact, and backup on a regular schedule. For more information, visit our web site at [www.fmsinc.com](http://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. From time to time you may find it necessary to re-assign the procedure to another user. Perhaps the original name was entered in error, or the author of the code no longer works for the organization. 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 5: 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**

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

- **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 3.5 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]. For more information, see page 38.

### **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 62.

---

## 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 50.

## 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 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**  
Disallow other users from editing or deleting the topic, and only allow 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 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.



**Tip**

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.

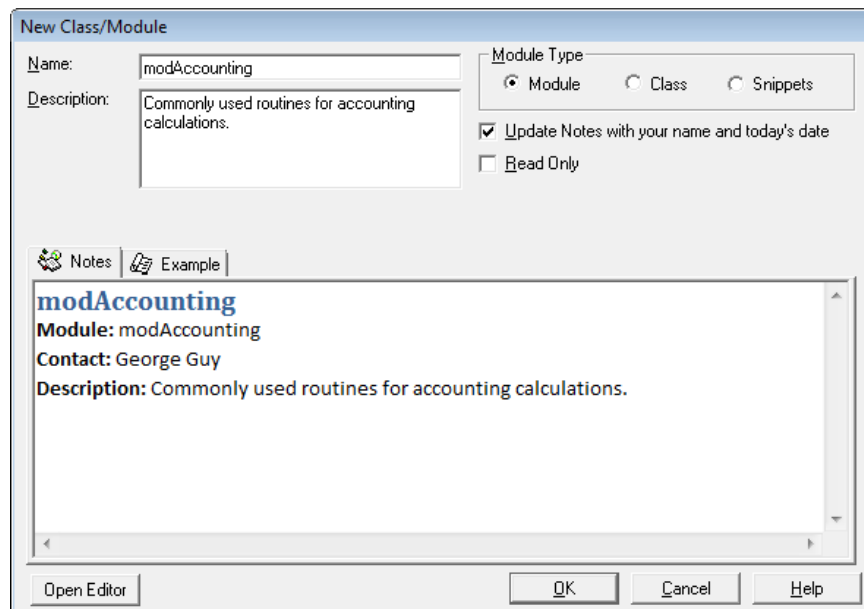
- **Read Only**

Disallow other users from editing or deleting the category, and only allow 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:



The screenshot shows a dialog box titled "New Class/Module". It has two input fields: "Name:" with the text "modAccounting" and "Description:" with the text "Commonly used routines for accounting calculations.". To the right, there is a "Module Type" section with three radio buttons: "Module" (selected), "Class", and "Snippets". Below that are two checkboxes: "Update Notes with your name and today's date" (checked) and "Read Only" (unchecked). At the bottom, there are three buttons: "Open Editor", "OK", "Cancel", and "Help". A preview area in the center shows the following text: "modAccounting", "Module: modAccounting", "Contact: George Guy", and "Description: Commonly used routines for accounting calculations.".

*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 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.

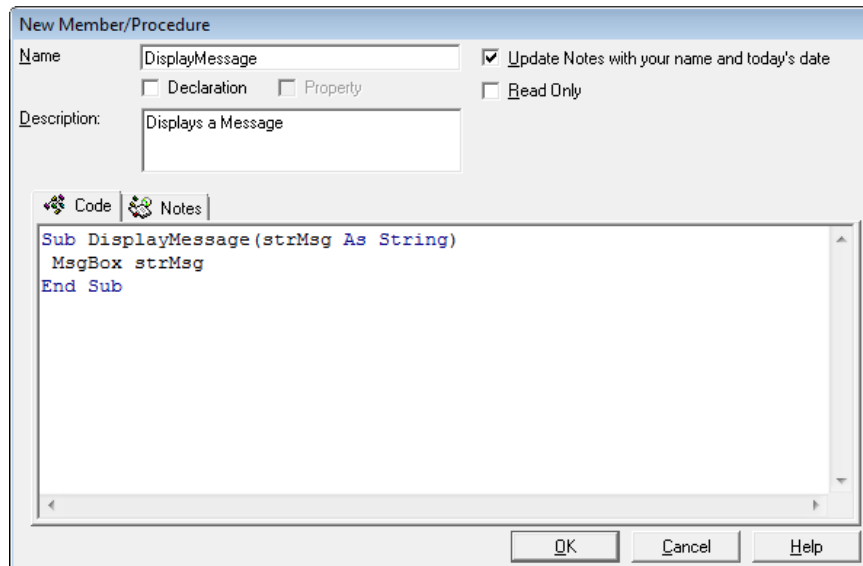
- **Read Only**

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  [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 marks 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 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 6: Source Code Exchange**.

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

The dialog box is titled "Import Class or Module". It features a section "Import From:" with two radio buttons: "File" (selected) and "Project". Each radio button is followed by a text input field and a browse button "...". Below this is the "Import into Category" section with a dropdown menu currently showing "Access, Reports". The "New Item Name:" section has a text input field containing "modNew". The "Create As:" section has two radio buttons: "Class" and "Module" (selected). At the bottom of the dialog are four buttons: "Preview", "OK", "Cancel", and "Help".

*Import Dialog*

### Import From

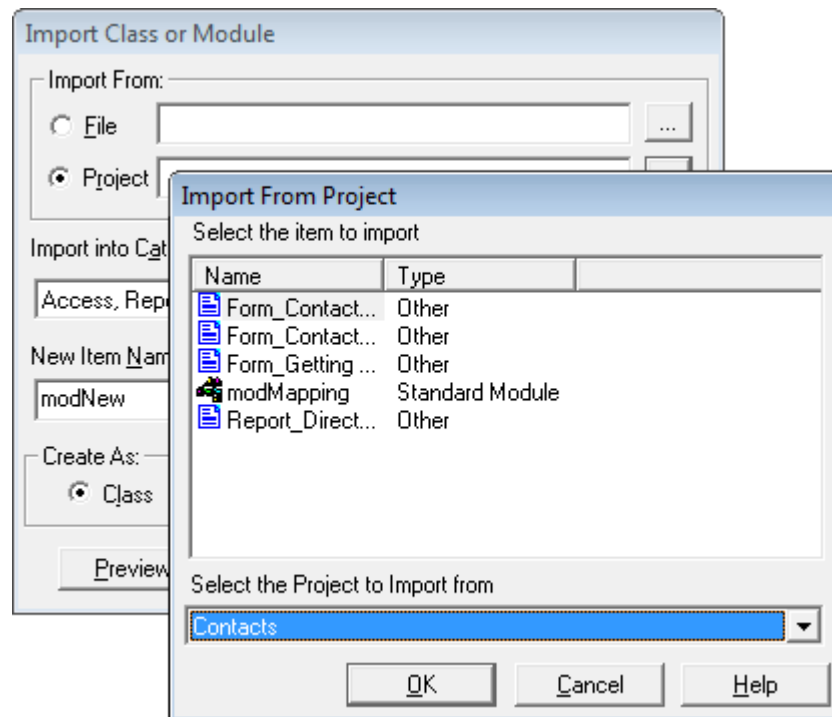
First, specify the source of the item you are importing:

#### ***File***

This can be any file, but only code information from the file is imported. See **How Non-Code Data Is Imported** on page 79 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** 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.

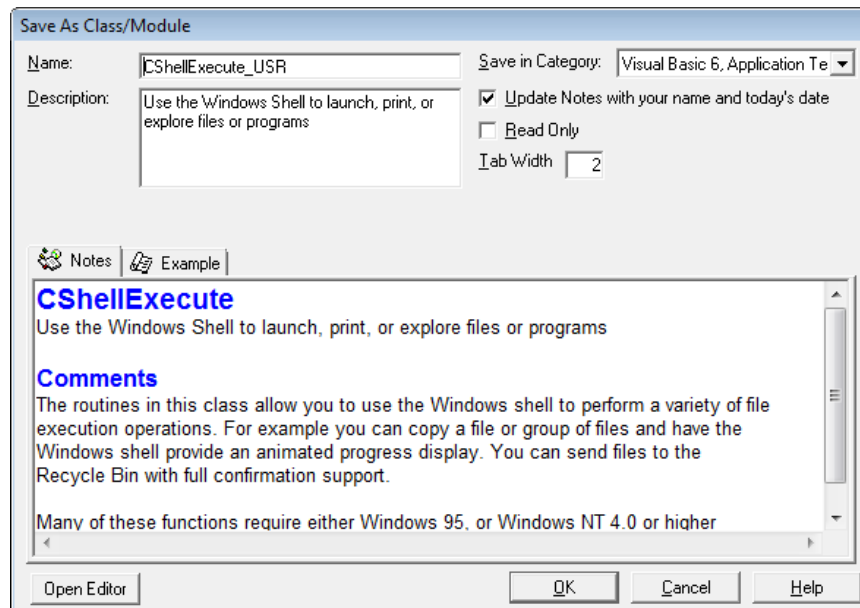
---

## 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 56).

### 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

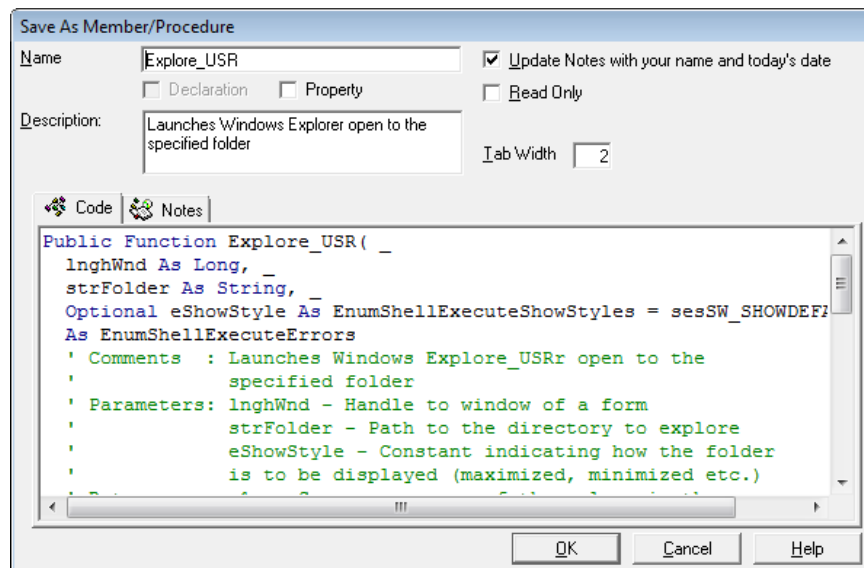
All other options and entries on this dialog are the same as when you create a new class or module. See Defaults for these options are stored under Tools, Options (on the Other tab).

Adding a New Class or Module on page 73 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 56).

The same rules for names and conflicts that applied to creating a new procedure also apply when you use the Save As function. See **Defaults for these options are stored** under Tools, Options (on the Other tab).

Adding a New Member or Procedure on page 75 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 75 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. If you need to change the ownership of a topic after it is created, use the User Name Replace option (see page 62).


### **Editing a Topic**

Select the topic to edit, and click the [Edit Topic] toolbar button  or select Edit, Topic from the menu. The options on this form are the same as those for the New Topic form. See page 70 for more information.

### **Editing a Category**

Select the category to edit, and click the [Edit Category] toolbar button , or select Edit, Category from the menu. The options on this form are the same as those for the New Category form. See page 70 for more information.

### **Editing a Class or Module**


Select the class or module to edit, and click the [Edit Class or Module] toolbar button  or select Edit, Class or Module from the menu.

The options are similar to those for the New Class or Module form, explained on page 73, except that you cannot change the Module Type option. For example, if you create the item as a Class, you cannot change it to a standard Module.

### ***Editing FMS Classes and Modules***

While you cannot modify FMS code, you can create your own copy of FMS classes and modules to customize. Use the Save As function to save the item under another name (see page 80), and edit your copy of the code.

### **Editing a Member or Procedure**

Select the procedure you wish to edit, and click the [Edit Member or Procedure] toolbar button  or select Edit, Member/Procedure from the menu. The options on this form are the same as those for the New Member or Procedure form, explained on page 75.


### ***Editing FMS Members and Procedures***

While you cannot modify FMS code, you can create your own copy of FMS procedures to customize. Use the Save As function to save the FMS procedure under another name (see page 80), and edit your copy of the code.

---

## **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 68.

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 6: 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:

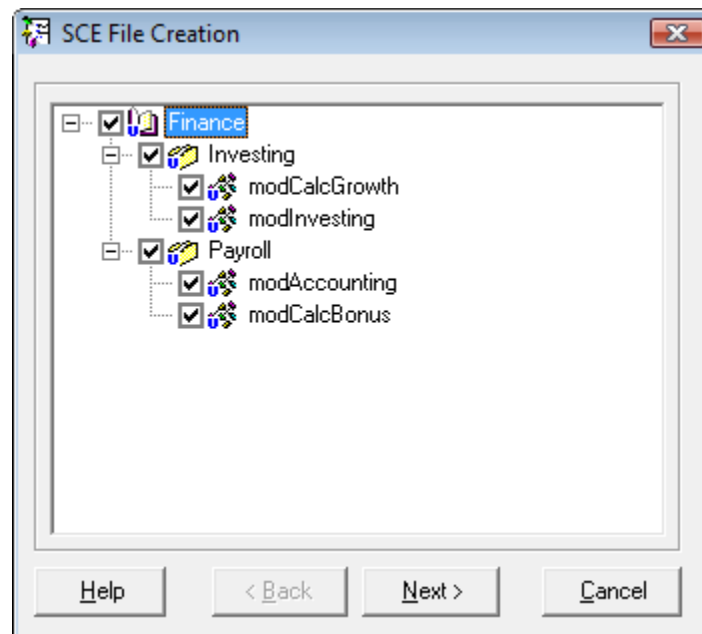
- Send updated code to offsite developers.
- Share new techniques with your programming team.
- Create subject-matter specific code for easy inclusion in a project.
- Post your own code on the web.

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:

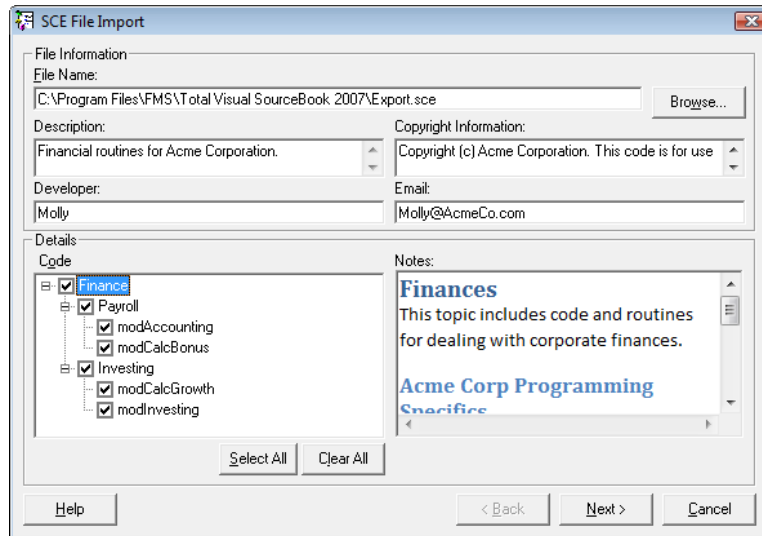
*SCE Export Descriptive Fields*

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 display create the file.

## Importing an SCE Package

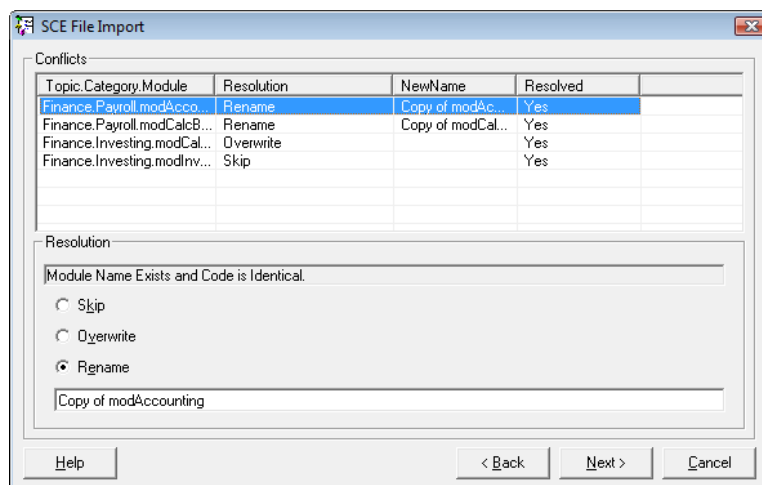
To import a Source Code Exchange (SCE) from another user, select File, Load SCE File to display the SCE import dialog:



*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 3.51 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 3.51 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**

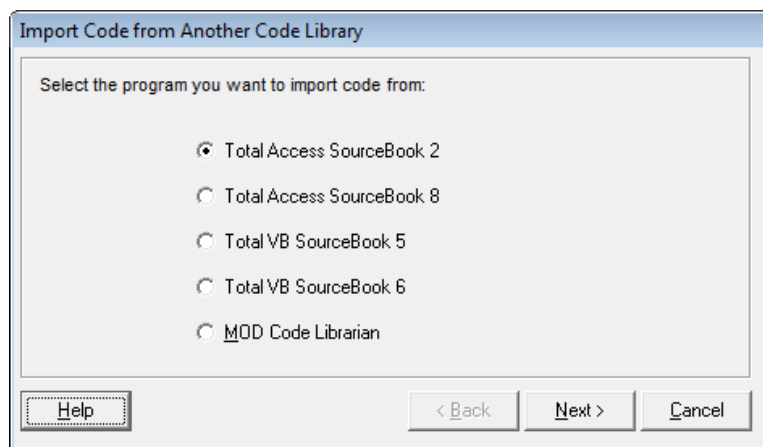
Once the database is open, select Tools, Database Utilities, Convert Database, To Prior Access Database Version from the Access menu. The Access Save File dialog appears asking you where you want to save the converted copy of the database.

### **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.



Tip

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 data 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 7: 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 help you resolve issues you may encounter. 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), newsgroups, information on new versions, betas, and other resources.

### Home Page

The FMS home page is located at:

`www.fmsinc.com`

News and important announcements are posted here.

### Support Site

The main support page is located at:

`www.fmsinc.com/support`

From this page, you can quickly locate the other support resources.

### Product Updates

FMS takes product quality very seriously. When bugs are reported and we can fix them, we make the updates available on our web site. If you are encountering problems with our product, make sure you are using the latest version.

Product updates can also be checked using the update wizard. See **Using the Update Wizard** on page 17 for details.

## Frequently Asked Questions (FAQs)

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

## Newsgroups

FMS also has general and product specific newsgroups. Connect with FMS Technical Support and other users there. Share your experiences, learn from others, and ask your questions in our virtual community:

[www.fmsinc.com/support/newsgrp.htm](http://www.fmsinc.com/support/newsgrp.htm)

Or visit our web site for additional information.

## Microsoft Patches

Our support site also includes links to Microsoft patches that are related to our products. Make sure you're using the latest versions by checking here or visiting the Microsoft site.

---

## Technical Support Options

FMS is committed to providing professional support for all of our products. We offer free access to our online FAQs and newsgroups. Bug reports, feature requests, suggestions, and general pre-sales questions related to our products are always available at no cost.

Additional maintenance plans are available to provide subscribers with enhanced technical support. This is the best way for you to stay current with the rapidly changing technologies that impact project development, and to ensure you are getting the maximum return from your software investment. Please visit our web site, [www.fmsinc.com](http://www.fmsinc.com), for the most up-to-date information.

Features & Benefits	Premium	Incident	Standard
Access to FAQs	✓	✓	✓
Access to Newsgroups	✓	✓	✓
Minor Upgrades/ Bug Fixes	✓	✓	✓
Telephone Support	✓	Per incident	First 30 Days
Email Support	✓	Per incident	First 30 Days

<b>Priority Response Time <sup>1</sup></b>	✓	✓	
<b>Senior Engineer Support Team</b>	✓	✓	
<b>Email Project for Testing</b>	✓	✓	
<b>Programmatic Code Assistance <sup>2</sup></b>	✓	✓	
<b>Major Upgrades for Current Version (not between Access versions)</b>	✓	Additional fee	Additional fee
<b>Cost</b>	Annual Fee	Fee Per Incident	Included
<ol style="list-style-type: none"> <li>1. Response generally within two business days. Actual resolution may take longer depending on complexity of the issue reported.</li> <li>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.</li> </ol>			

### 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.

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 and use a subscription. Please ensure that 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 newsgroups, and responses to bug reports and feature requests for that version.

Please note that the person requesting support must also be the registered user of the product. Registration is required and will be requested by our Technical Support professionals.

---

## **Contacting Technical Support**

If the troubleshooting suggestions and other support resources fail to resolve your problem, please contact our technical support department. We are very interested in making sure you are satisfied with our product.

### **Registering Your Software**

Please register your copy of Total Visual SourceBook. 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.

You may register your software by mailing in your registration card or directly on our web site.

### **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 via email at:

`Support@fmsinc.com`

Please provide detailed information about the problem that you are encountering. This should include the name and version of the product, your operating system, and the specific problem. If the product generated an error file, please submit that as well.

With email, technical support issues can be more accurately resolved and tracked in our internal technical support system. Email also gives us more time to understand the entire problem and allows our technical support staff to contact the developers with the entire story when necessary. Please bear in mind that a unique issue may involve meetings between the technical support staff and product developers, so your patience is appreciated.

### **Microsoft Technical Support**

FMS only provides technical support for its products. If you have questions regarding Microsoft products, please contact Microsoft technical support.

# Index

## #

#M#, 63  
#P#, 63  
#X#, 63

## A

Add-In, 34  
adding items, 70–77  
    new category, 71  
    new class, 73  
    new member, 75  
    new module, 73  
    new procedure, 75  
    new topic, 70

## B

backups, 66  
blog, 9  
bookmarks, 41, 46, 54–55  
    creating, 54  
    deleting, 55  
    editing, 55  
by reference, 29  
by value, 29

## C

class, 29  
class dependencies, 23  
code conflicts, 24  
code dependencies, 22–23  
Code Explorer, 13, 33–64  
    bookmarks tab, 46  
    code tab, 44

    details tab, 45  
    example tab, 45  
    export, 47  
    information window, 44  
    notes tab, 45  
    opening, 34  
    search results tab, 46  
    toolbar, 38  
    tracking window, 46  
    treeview, 42  
code group, 13, 14, 35  
    join, 14, 15, 16, 36  
    location, 14  
    new, 16  
    options, 59  
    shared, 14, 15  
Code Librarian, 89  
Code Repository, 5, 65–90  
    adding items, 70  
    delete code, 83  
    editing, 82–83  
    import code, 77–80  
    making backups, 66  
    overview, 66  
commenting, 29  
compact, 63  
conflicts, 24  
consistency, 22  
conventions, 24–26  
    controls, 24–26  
    scope, 24–26  
    variables, 24–26  
copy code, 39, 47  
create new code group, 16  
created by, 69  
creation date, 69

## D

delete code, 83

dependencies, 22, 23, 47, 48, 70  
distribution rights, i, ii

## E

edit, 82–83  
  category, 82  
  class, 82  
  member, 83  
  module, 82  
  procedure, 83  
  topic, 82  
editing conflicts, 67  
error handling, 26–27, 60–63  
  tokens, 63  
examples, 30  
Explorer. *see* Code Explorer  
export code, 39, 47  
  clipboard, 48  
  file, 49  
  notepad, 49  
  project, 49  
export options, 57

## F

FMS web site, 8–9, 94

## H

highlights, 4

## I

import, 77–80  
  SCE file, 88  
information window, 44  
installation, 11–19  
  multi-user, 15, 16  
  requirements, 12  
  single user, 15, 16  
  upgrading, 12  
Internet web site, 8–9

## J

join code group, 14, 15, 16, 36

## L

licensing, i–iii  
login, 35

## M

menu, 37  
Microsoft Office Developer Code Librarian, 89  
modification date, 70  
modified by, 70  
module, 29  
multi-user, 14, 66, 67  
  installation, 15, 16

## N

naming conventions, 24–26  
  controls, 24–26  
  scope, 24–26  
  variables, 24–26  
newsgroups, 9, 95  
notes  
  updating, 58

## O

options, 55–63  
  code group, 59  
  error handling, 60–63  
  export, 57  
  fonts, 58  
  syntax highlighting, 55  
  tab width, 57  
  treeview, 59  
  window size, 58

## P

print, 41, 50  
programming standards, 22, 28  
  consistency, 22  
  naming conventions, 24–26  
publishing library code, ii

## R

read only, 59, 68, 75  
readme file, 94

redistribution rights, i  
refresh, 38, 68  
registration, 99  
repair, 63  
replace user name, 64  
Repository. *see* Code Repository, *see* Code Repository  
Rich Text Editor, 71, 72, 74, 77

## S

save as, 40, 57, 80, 81, 83  
search, 6, 39, 51–53  
    boolean operators, 52  
    category, 52  
    filters, 53  
    options, 52  
    results, 53  
    results tab, 46  
    search phrase, 52  
service packs, 28  
single user installation, 15, 16  
Source Code Exchange (SCE), 85–92  
    creating SCE files, 86–87  
    importing SCE files, 88  
standalone, 34  
standards, 22, 28  
    consistency, 22  
    naming conventions, 24–26  
starting Total Visual SourceBook, 34  
status bar, 38  
storing code, 66  
support, ii, 93–100  
syntax highlighting, 55  
system requirements, 12

## T

tab width, 57  
technical support, ii, 93–100, 99  
    Microsoft, 100  
tokens, 63  
toolbar, 37  
    icons, 38–41  
Total Access Memo, 71, 72, 74, 77  
Total Access SourceBook, 7, 12  
Total VB SourceBook, 7, 12  
Total Visual SourceBook 2000, 7, 12  
tracking window, 46  
treeview, 42  
troubleshooting, 94  
TVSBIC90.MDB, 13, 19, 60  
TVSBUC90.MDB, 13, 19, 60

## U

un-installing, 18  
update wizard, 18  
updates, 8, 28  
upgrading, 12  
user name replace, 64

## V

Visual Basic project, 49

## W

web site, 8–9, 94  
Windows API, 26