Contents

[Phone](#)

[Credit Card](#)

[SWREG](#)

[CHECK](#)

[Shipping and Handling](#)

Technical Support

[CompuServe](#)

[America On-Line](#)

[Fax](#)

[Mail](#)

[Voice](#)

Introduction

Congratulations! You have downloaded a <u>shareware</u> copy (or purchased a registered copy) of <u>SCOPER</u>, an essential add-on tool for serious <u>VB</u> developers.

SCOPER's primary job is to examine VB source code and find elements that are wasting space but are never used.

SCOPER will find variables, constants, function and subroutine <u>declarations</u>, and function and subroutine code that are unreferenced throughout your <u>project</u>.

Related Topics:

<u>Eliminate Dead Code</u>

<u>Produce Project Cross-Reference Reports</u>

<u>Unveil Project Statistics</u>

<u>Project Statistics Drill Down</u>

<u>System Requirements</u>

Eliminate Dead Code

SCOPER (the registered version) will automatically rewrite your project source code (in a different directory) eliminating all unused elements from your project. The shareware version will produce a report that shareware users can use to manually eliminate unused elements from their projects.

Produce Project Cross-Reference Reports

SCOPER (the registered version) will produce an optional cross reference report for the entire project, showing all references for project elements. The report can be "filtered" by "scope" and "type". For example, the SCOPER cross-reference filter can be used to limit the report to "module" scope "constants".

Unveil Project Statistics

SCOPER produces a statistical summary for each project file, and overall totals for the entire project. This report can be used to see, for example, just how many controls are on XYZ.FRM. Any file on the statistical summary can be "double-clicked" to "drill-down" to a more detailed statistical level.

Project Statistics Drill Down

Double-clicking on a file on the statistical summary screen will bring up the "drill-down" statistics screen for that <u>project</u>. This shows a matrix of counts of element types, by <u>scope</u>, total and used. The <u>drill down</u> data can be used to see, for example, exactly how many global strings are defined in FOO.BAS, and how many are unused.

System Requirements

SCOPER requires that Visual Basic 3.0 Professional Edition for Windows be installed on the same machine.

SCOPER requires that all project source code be "saved as text". The registered version will perform an "in-flight" conversion to text if any binary source is found.

SCOPER   B.xx will work under Windows 3.1, WFWG 3.11, Windows NT 3.5 and 3.51, and Windows 95.

SCOPER requires at least 1 meg of disk space. Up to 5 megs of temporary work space can be needed when SCOPER is cross-referencing a large project.

As for processor speed and RAM, it boils down to this. If you can run VB on the machine, you can certainly run SCOPER.

The Distribution Diskette

SCOPER does not come with a SETUP.EXE program. It is intended to be used by serious VB programmers who already are using a wealth of VBX and DLL add-ons. We will not risk overlaying one of your tools with a similar (but perhaps older) tool with the same name.

Related Topics:

How To Begin

Copying the Files

Making a SCOPER Program Manager Group

README.TXT and OVERVIEW.TXT

Ready to Roll!

Starting SCOPER

How To Begin

Make a directory with any name you want. One popular method is to make a directory called SCOPER and a sub-directory called SCOP_xxx where xxx = the version of SCOPER you are installing. For example, if you are installing version B.06, you would make a directory called SCOPER and a sub-directory called SCOP_B06.

Copying the Files

Copy the contents of the distribution diskette into the directory you made. <u>SCOPER</u> needs the following files to function properly:

SCOPER.EXE

SCOPER.HLP

SPREAD20.VBX

QPRO200.DLL

If you are a <u>shareware</u> user, unzip the SCOPER file into the directory you created.

Making a SCOPER Program Manager Group

Regular SCOPER users normally make a SCOPER Program Manager Group. In the group, add the SCOPER.EXE program and the SCOPER.HLP on-line help file.

README.TXT and OVERVIEW.TXT

The distribution diskette contains a file called README.TXT. We recommend that you print this file and keep it handy. It contains last-minute documentation that may not be included in this manual. The OVERVIEW.TXT file is just a copy of the SCOPER description seen on various BBS services throughout the world.
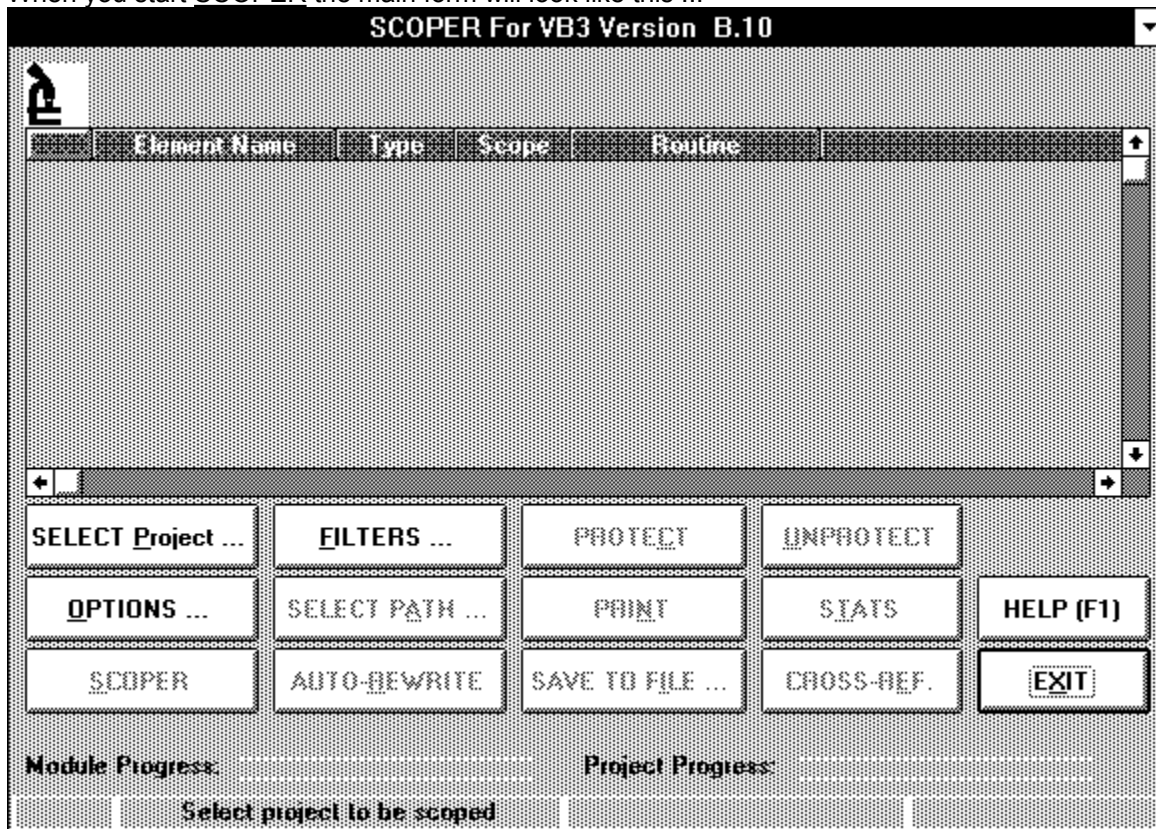
Ready to Roll!

That's it! SCOPER's installed and you're ready to begin tuning your VB projects into the smallest, fastest .EXE's possible!

Starting SCOPER

Assuming you created a <u>SCOPER</u> group as described above, simply double-<u>click</u> the SCOPER icon (the small curved microscope) and you're underway. The rest of this manual describes the various forms you will see in SCOPER and provides tips on how to get the most out of the product. WELCOME ABOARD!

Startup

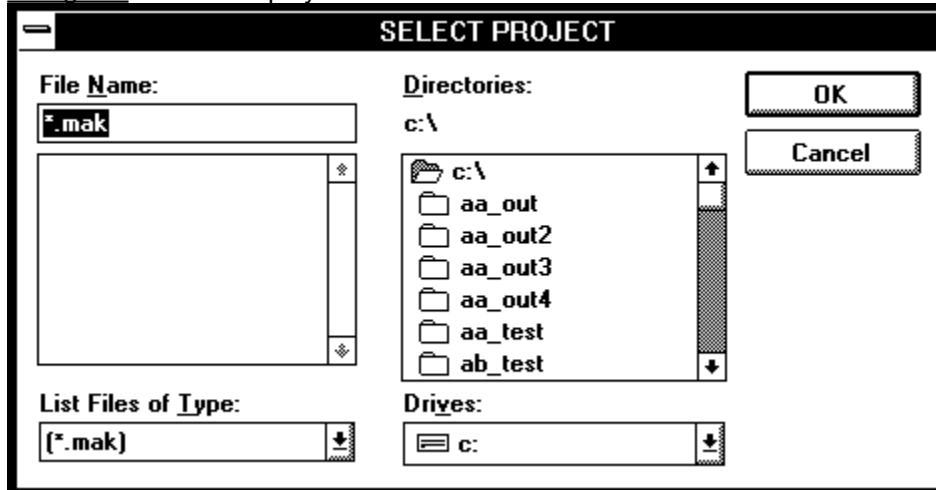When you start SCOPER the main form will look like this ...

Notice that the only enabled buttons are 'SELECT PROJECT ...', 'OPTIONS ...,' 'FILTERS ...' 'HELP', and 'EXIT'. Click 'SELECT PROJECT' to choose the VB project you want SCOPER to analyze.

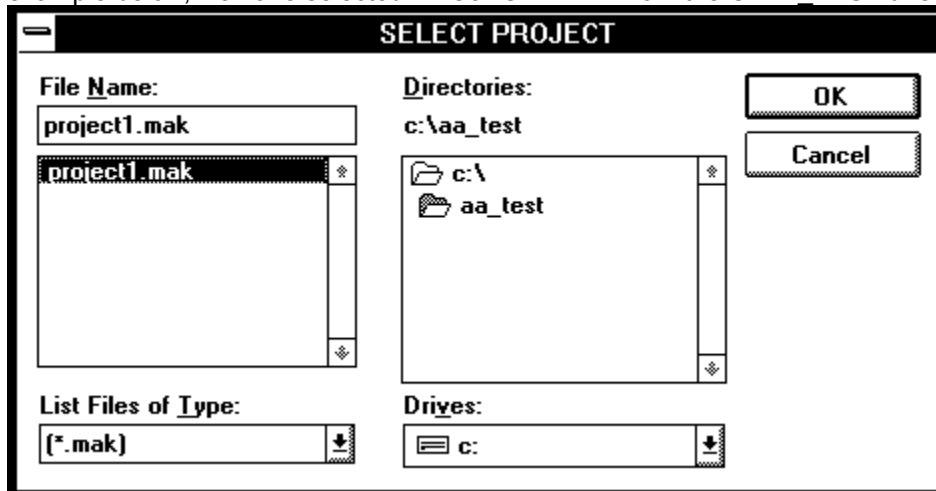Notice also that "hot-keys" are provided for all SCOPER functions.

Selecting a Project

When you click the 'SELECT PROJECT' button on the main SCOPER form, the SELECT PROJECT dialog box will   be displayed as shown below:

Use standard Windows methods to select the drive, path and project .MAK file you want analyzed. In the example below, we have selected PROJECT1.MAK from the C:\AA_TEST directory:
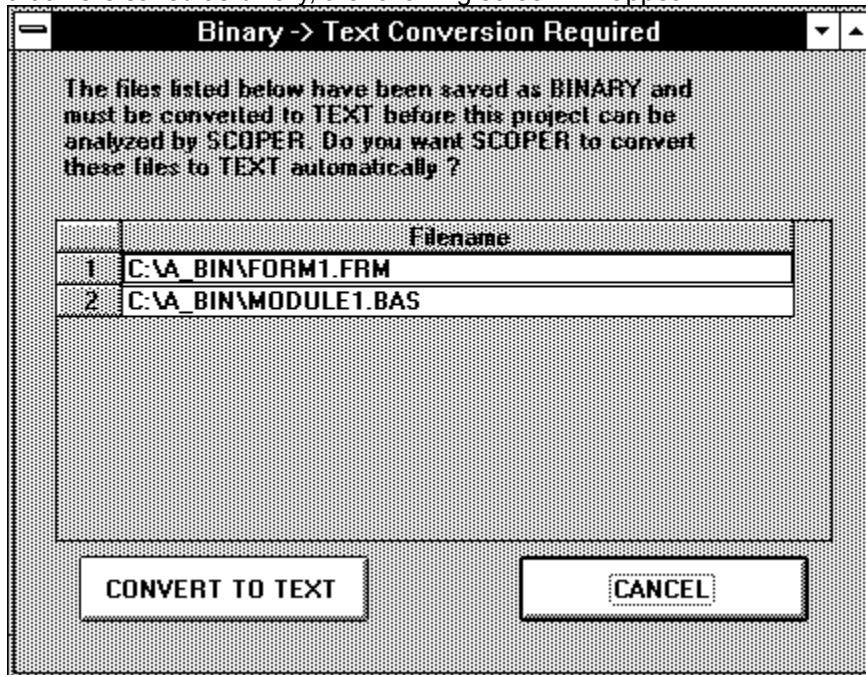

Once you have selected the project .MAK file, click OK.

Related Topics:

In-Flight Binary -> Text Conversion
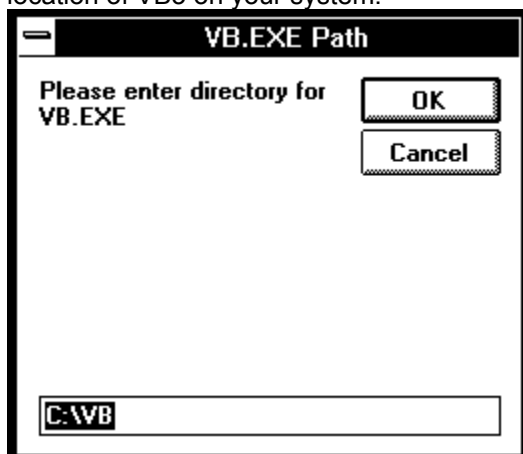
In-Flight Binary -> Text Conversion

THIS FEATURE IS AVAILABLE TO REGISTERED USERS ONLY.

SCOPER requires that all project source modules be 'saved as text'. If it finds any project source modules that were saved as binary, the following screen will appear:
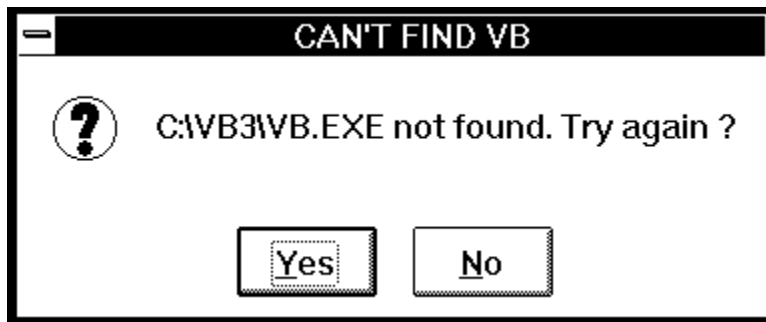

The modules listed in the spreadsheet on the "Binary->Text Conversion Required" form are currently in "binary" format. Click on "CONVERT TO TEXT" to invoke SCOPER's in-flight binary->text conversion system.

The first time you use this feature, SCOPER will display the following form, asking you to identify the location of VB3 on your system:
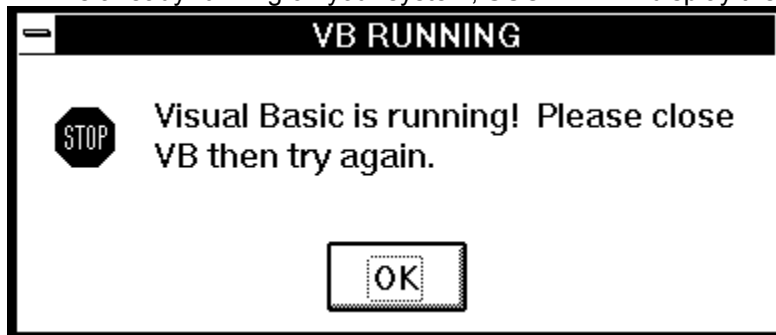

Once SCOPER has located VB.EXE, it executes VB and passes it a series of keystrokes to convert modules that have been saved as binary to text.

NOTE: If SCOPER cannot find VB.EXE in the specified path, it will display the following message:
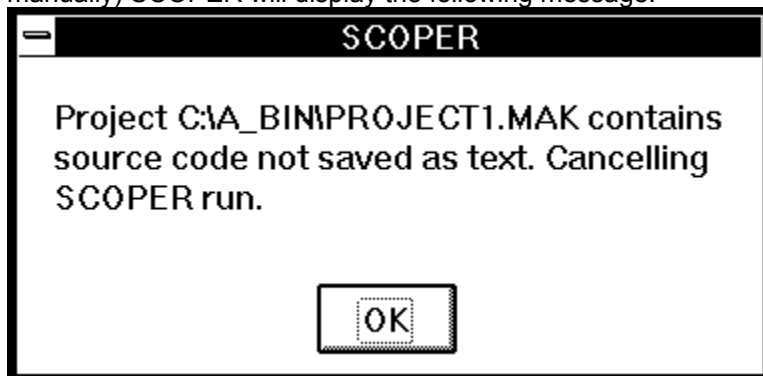
```
┌──────────────────────────────────────────────┐
│ ▬            CAN'T FIND VB                     │
├──────────────────────────────────────────────┤
│                                                │
│   ╭──╮                                         │
│  ( ? )   C:\VB3\VB.EXE not found. Try again ?  │
│   ╰──╯                                         │
│                                                │
│        ┌─────────┐      ┌─────────┐            │
│        │  Yes    │      │  No     │            │
│        └─────────┘      └─────────┘            │
│                                                │
└──────────────────────────────────────────────┘
```

Click YES to change the path name where VB.EXE exists.

If VB is already running on your system, SCOPER will display the following message:

```
┌──────────────────────────────────────────────┐
│ ▬             VB RUNNING                       │
├──────────────────────────────────────────────┤
│                                                │
│  ╭────╮   Visual Basic is running!  Please close│
│  │STOP│   VB then try again.                   │
│  ╰────╯                                        │
│                                                │
│              ┌─────────┐                       │
│              │   OK    │                       │
│              └─────────┘                       │
│                                                │
└──────────────────────────────────────────────┘
```

You will have to shut down VB and then start the in-flight binary-text conversion process again.

If you are using an evaluation copy of   SCOPER (or if you want to convert your binary modules to text manually) SCOPER will display the following message:

```
┌──────────────────────────────────────────────┐
│ ▬               SCOPER                         │
├──────────────────────────────────────────────┤
│                                                │
│  Project C:\A_BIN\PROJECT1.MAK contains        │
│  source code not saved as text. Cancelling     │
│  SCOPER run.                                   │
│                                                │
│              ┌─────────┐                       │
│              │   OK    │                       │
│              └─────────┘                       │
│                                                │
└──────────────────────────────────────────────┘
```
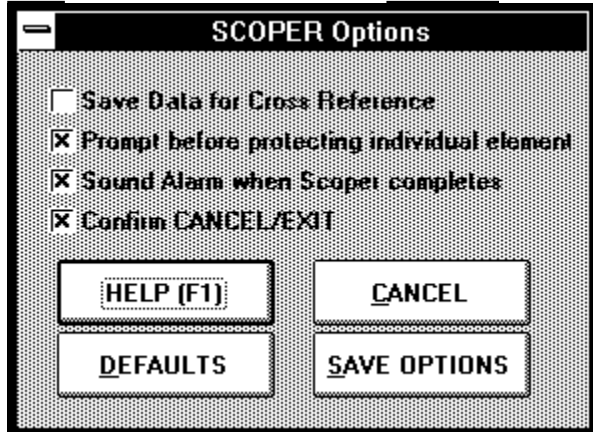
You will have to re-save the binary module(s) 'as text' in VB before running the project through SCOPER. Remember to use the "Save As" function with each individual module in VB to convert from binary to text.

Assuming all project forms and modules are now in text format, the main SCOPER form will now be displayed looking like this:

SCOPER For VB3 Version B.10

C:\AA_TEST\PROJECT1.MAK

Element Name	Type	Scope	Routine	

SELECT Project ...   FILTERS ...   PROTECT   UNPROTECT

OPTIONS ...   SELECT PATH ...   PRINT   STATS   HELP (F1)

SCOPER   AUTO-REWRITE   SAVE TO FILE ...   CROSS-REF.   EXIT

Module Progress: Project Progress:

Click "SCOPER" to begin

Notice that the name of the project is displayed at the top of the form. Also, the SCOPER button is now enabled and the message 'Click SCOPER to begin' is displayed in the message panel. Before beginning, however, we will make sure that the SCOPER 'OPTIONS ... ' and 'FILTERS...' are the ones we want activated while the project is being analyzed.

SCOPER Options

Click 'OPTIONS ...' on the main SCOPER form to bring up the options form ...

The options shown above are the SCOPER option 'defaults'. Varying these option settings permit you to customize the way SCOPER analyzes your source code based upon your individual needs. Details on customizing options are given in the sections below.

Related Topics:

Save Data for Cross-Ref.

Prompt Before Protecting Element

Sound Alarm When SCOPER Completes

Confirm CANCEL/EXIT

Save Data for Cross-Ref.

SCOPER comes with a built-in cross-referencing engine that can be activated by checking this box. If cross-referencing is active while the project is being analyzed the 'CROSS-REF' button will be enabled when the analysis is complete, and you will be able to view, print, and save a full or custom filtered cross-reference report for your project.

That's the good news. The bad news is that cross-referencing will increase the analysis run-time by about 40%. We recommend leaving this check off (the default state) until you have completed (or nearly completed) the development cycle of the project.

Prompt Before Protecting Element

When the <u>analysis phase</u> completes, <u>unused elements</u> are displayed in the grid on the main <u>SCOPER</u> form. All elements appearing in this grid will be deleted during <u>auto-rewrite</u>. If you see an individual element that you want to save (i.e., have SCOPER leave it alone) you can <u>click</u> on the element's row in the grid and then click the PROTECT button.

SCOPER will prompt you for confirmation before protecting the element if this option is on.

Note that the same effect can be achieved by double-clicking the element's row in the grid.

Sound Alarm When SCOPER Completes

SCOPER runs in cooperative multi-tasking mode during the analysis phase. This means that you are able to do other tasks on your computer while SCOPER is running even if you are using a non-preemptive environment such as Windows 3.1 or WFWG 3.11. When this box is checked, SCOPER will sound an alarm when the analysis phase completes.

Confirm CANCEL/EXIT

When this box is checked, <u>SCOPER</u> will display a "confirmation" message box when the CANCEL button is clicked during the <u>analysis phase</u> or when EXIT is clicked on the main SCOPER form.

SCOPER Filters

The checkbox array on the Filters screen can be used to have <u>SCOPER</u> ignore <u>unused elements</u> of any type/<u>scope</u> combination during the <u>analysis phase</u>.

For example, suppose you are developing a set of <u>VB</u> functions and subroutines that will be used by programmers in other projects. You would not want SCOPER to identify these procedures as 'unused' and get rid of them during <u>auto-rewrite</u>. You can turn off function and subroutine checking by removing the appropriate checks as follows:

Another common use of the <u>filter</u> screen is to prevent SCOPER from identifying unused elements in CONSTANT.TXT or WIN30API.TXT. Suppose you have completed only 50% of the code in your <u>project</u>. You want to have the project analyzed by SCOPER but you don't want the unused element report cluttered up by thousands of unused global constants and <u>declarations</u> in CONSTANT.TXT or WIN30API.TXT. Removing the global constant and declaration checks does the trick as follows:

Eliminate Unused Element Filters

TYPE	LOCAL	MODULE	GLOBAL
Integer	☒	☒	☒
Long	☒	☒	☒
Float	☒	☒	☒
Double	☒	☒	☒
Currency	☒	☒	☒
String	☒	☒	☒
Variant	☒	☒	☒
User Type	☒	☒	☒
Const	☒	☒	☐
Declare		☒	☐
Function		☒	☒
Sub		☒	☒

X = Unused elements of this TYPE/SCOPE combination will be displayed in the unused element grid and will be REMOVED during auto-rewrite. Removing check will suppress display in the grid and will PROTECT these elements during auto-rewrite.

[HELP (F1)] [CANCEL]

[DEFAULTS] [SET OPTIONS]

As you can see, the Filters form gives you complete control over the elements SCOPER will protect during auto-rewrite.

The DEFAULTS button will reset all checks to their default states.

The CANCEL button will return you to the main SCOPER form without saving any changes you have made to the checkbox states.

The SET OPTIONS button will activate the options and filters you have selected.

Be aware that you can enter the Filters form from the main SCOPER form and modify filter settings *any time you want*. If you have run a project through the analysis phase and forgot to set some filters you can go back to the Filters form *after* the analysis phase is done. SCOPER will re-load the unused element spreadsheet based upon your new filter settings *without* forcing you to run the project through the analysis phase again.

Also, remember that option and filter settings are *not* statically held across SCOPER sessions. Each time you start SCOPER, the option and filter settings revert to their default state. This is done by design because most of the time the default settings should be exactly the ones you want. Options and filters are modified only in relatively unusual situations.

Starting the Analysis Phase

After you have selected the project to be analyzed, the SCOPER main form will look like this:
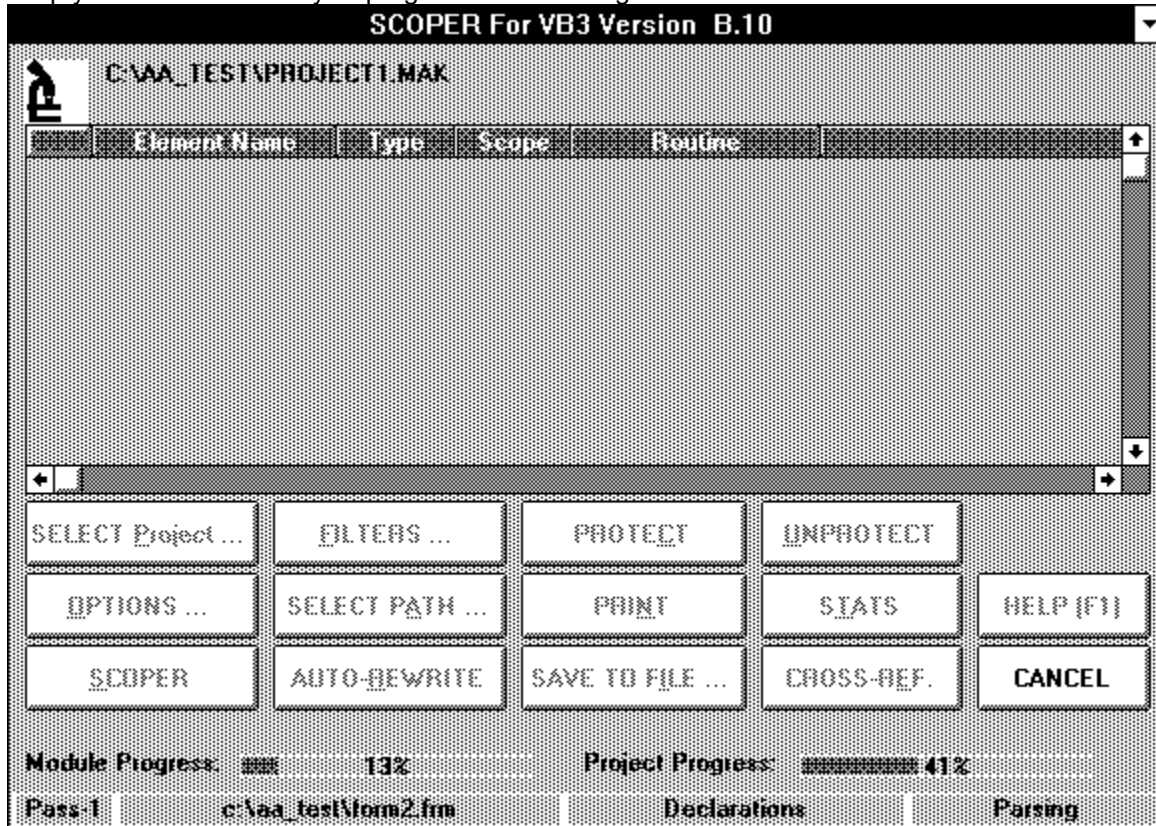

After checking the SCOPER options form to make sure the option settings you want to have in effect during the run are correct, click SCOPER to begin the analysis.

Related Topics:

Analysis Progress Displays

Analysis Progress Displays

While your project is being analyzed by SCOPER, the message bar at the bottom of the main form will keep you aware of the anlysis progress. The message bar is divided into four sections:

Starting from the left, the first message bar section tells you what 'pass ' SCOPER is in. SCOPER makes two passes through your source code during the analysis phase. In the above example, SCOPER is in Pass-1.

The second section displays the project source file currently being analyzed. In the above example the form module C:\AA_TEST\FORM2.FRM is being processed.

The third section contains the section of the source module currently being examined. In the above example SCOPER is processing the Declarations section of C:\AA_TEST\FORM2.FRM.

The fourth section tells you what 'action' is currently underway. In the above example, SCOPER is 'Parsing' the source code in the Declarations section of C:\AA_TEST\FORM2.FRM.

Directly above the message bar are two progress bars labeled 'Module Progress' and 'Project Progress'.

The Module Progress bar graphically displays analysis progress for the file currently being examined. In the above example, SCOPER has completed analyzing 13%   of the source code in module C:\AA_TEST\FORM2.FRM.

The Project Progress bar graphically displays analysis progress for the project currently being examined. In the above example, SCOPER has completed analyzing 41%   of the source code in the C:\AA_TEST\PROJECT1.MAK project.

Notice also that the only button active during the analysis phase is CANCEL. You can cancel a SCOPER

run at any time. If you have left the "Confirm CANCEL/EXIT" option on, you will be prompted to confirm the analysis phase cancellation request.

Because SCOPER operates in cooperative multi-tasking mode, feel free to minimize SCOPER and use your computer for other tasks while analyzing large projects. SCOPER will maximize itself when the analysis phase completes and sound an alarm if the "Sound Alarm When Scoper Completes" option is active.

When the Analysis Phase Completes

When SCOPER finishes analyzing your source code, the main form will look like this:

Related Topics:

The Unused Elements Spreadsheet

Enabled Buttons

The Message Bar

The Unused Elements Spreadsheet

The spreadsheet in the center of the main form contains all the unused elements found in the project according to the filters in effect during the analysis phase. The spreadsheet is sorted by filename, routine, and element name.

The first column contains the name of the element. The second column contains the element type. The third column contains the scope of the element. The fourth column contains the routine name where the element is defined. The fifth column contains the name of the source module where the element was found.

On row #3 of the spreadsheet in the above example, we see that SCOPER found an unused *Local Integer* named *pnLocalInt* in the *Form_Load* event in *C:\AA_TEST\FORM1.FRM*.

The vertical scrollbar on the right side of the spreadsheet can be used to scroll up and down to view other unused elements in the project. The horizontal scrollbar at the bottom of the spreadsheet can be used to scroll long filenames into view.

Notice that the FILTERS ... button is enabled. If you want to change the unused scope/type element protection filters you can bring up the Filters form *after* the analysis phase completes, change some filters and click SET OPTIONS on the Filters form. SCOPER will reload the unused element spreadsheet based upon your new option settings *without* re-analyzing the project. SCOPER is able to do this because *all* unused elements are held in memory. The unused elements you can see in the spreadsheet are only those that passed your custom unused element protection filters.

Enabled Buttons

When the <u>analysis phase</u> completes, the button states at the bottom of the main <u>SCOPER</u> screen will be as follows:

SELECT <u>PROJECT</u>	Enabled
OPTIONS	Enabled
FILTERS	Enabled
SCOPER	Disabled
SELECT <u>PATH</u>	Enabled
<u>AUTO-REWRITE</u>	Disabled
PROTECT/UNPROTECT	Enabled
PRINT	Enabled
SAVE TO FILE	Enabled
STATS	Enabled
CROSS-REF.	Disabled
HELP (F1)	Enabled
EXIT	Enabled

Each button is described in detail in the following sections.

The Message Bar

When SCOPER completes the analysis phase, the message bar will display 'Done' in the first section. The second section will display the total lines of source code in the project and the size of the project in bytes. The third section contains the elapsed time for the analysis phase in HH:MM:SS format. In the above example, we see that C:\AA_TEST\PROJECT1.MAK contained 148 lines of code totalling 2,644 bytes. It took SCOPER 6 seconds to analyze this project.

Run Times

How Long Does it Take ?

The time it takes for <u>SCOPER</u> to analyze a <u>project</u> depends upon several variables:

1. <u>Lines of Source</u>: Obviously, the bigger the project the longer it will take to analyze. Not so obvious is the fact that SCOPER will take longer to analyze a project with few very large modules than a project with lots of smaller modules. A good rule of thumb is to anticipate about 10 minutes of run time for every 15,000 lines (or 500k) of source code. This estimate is based upon a 486/66 processor with 8 megs of memory.

2. <u>Unused Elements:</u> The larger the number of unused elements, the longer it will take SCOPER to analyze the source. Projects that have CONSTANT.TXT, WINAPI30.TXT, etc. in the .<u>MAK</u> file and are using almost none of the thousands of <u>declarations</u> and constants in these modules will take somewhat longer to analyze.

3. <u>Global and Module Scope:</u> SCOPER will take slightly longer to analyze projects that use tons of Global and Module scope variables.

4. <u>Cross-Reference Data:</u> The 'SAVE DATA FOR CROSS-REF' option (available on the SCOPER OPTIONS form) will allow you to produce a beautiful and useful report and Excel-importable spreadsheet of your project's cross reference information. Be aware, however, that turning this option on will increase the SCOPER <u>analysis phase</u> run time by about 40%.

Protecting Individual Elements

The elements that appear in the unused element spreadsheet on the main SCOPER form will be removed from the project during the auto-rewrite process. If you want to protect a group of elements from removal, use the OPTIONS screen to identify the group you want to protect.

If you want to protect an individual element from deletion, use the PROTECT and UNPROTECT buttons as described below.
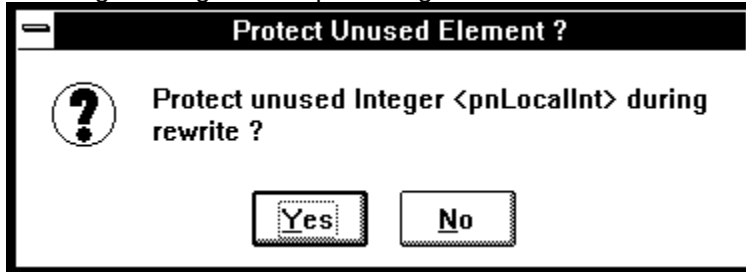
Related Topics:

The PROTECT Button

The UNPROTECT Button

The PROTECT Button

To protect an individual unused element from deletion, <u>click</u> on the element's row in the spreadsheet and then click the PROTECT button.

If the "Prompt Before Protecting Individual Elements" option is in effect, <u>SCOPER</u> will display the following message before protecting the element:

Click Yes to protect the element. You will see the element disappear from the unused element spreadsheet. This will cause SCOPER to leave the element alone during <u>auto-rewrite</u>.
Note that the same effect can be achieved by double-clicking on the element's row in the spreadsheet.

The UNPROTECT Button

If you change your mind and want to unprotect individual elements you have previously protected, click the UNPROTECT button. All elements you had previously individually protected will return to the unused element spreadsheet.

Overview

THIS FEATURE IS AVAILABLE TO REGISTERED USERS ONLY.

The <u>auto-rewrite</u> feature is, obviously, largely responsible for <u>SCOPER</u>'s popularity.

To new registered users, rest assured that SCOPER will NOT change a single line of your original source code. The first thing SCOPER does when auto-rewrite is invoked is to ask you to select a directory where SCOPER can put the rewritten source. If you try to point to the original <u>project directory</u>, SCOPER will ask you to select another <u>path</u>.

You don't have to wait until your project is nearing completion to take advantage of SCOPER. Many seasoned SCOPER users frequently analyze their code to create custom CONSTANT.TXT files, keep track of how large the project is getting, find out how many controls are on each form, etc. Frequent *scoping* during the development process gives you ongoing statistics about you project growth.

Related Topics:

<u>Recursion</u>

<u>Iteration</u>

Recursion

Although <u>recursion</u> is not a particularly popular technique in the <u>VB</u> development community, it needs to be addressed here.

If ROUTINE-A references itself but is not referenced anywhere else in the <u>project</u> outside of its own <u>scope</u>, <u>SCOPER</u> will identify it as unused and ROUTINE-A will be eliminated during <u>auto-rewrite</u>.

If ROUTINE-A calls ROUTINE-B, and ROUTINE-B calls ROUTINE-A SCOPER will not eliminate either routine because, technically speaking, both are referenced outside their own scope.
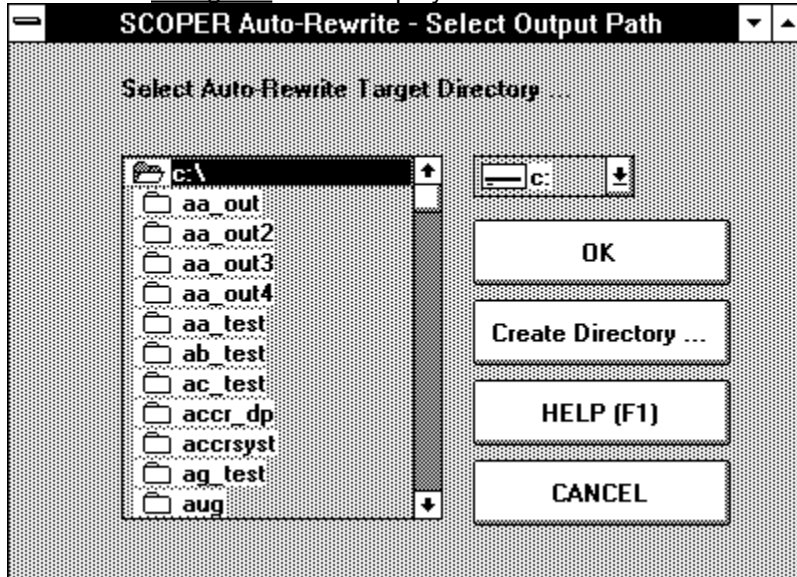
Iteration

Variables and constants that are only used in routines that are actually 'dead code' will not be eliminated by SCOPER during the first rewrite. If the rewritten project is analyzed and rewritten again, these variables and constants will be eliminated.

What all this means is that the best way to use auto-rewrite is to set up two directories to receive the new code. Put the first rewrite in directory-1. Run the .MAK from directory-1 through SCOPER and, if any unused elements still exist, rewrite it into directory-2. Then, run the .MAK from directory-2 through SCOPER. If any unused elements still exist, put this rewrite in directory-1. Repeat this process until the project is clean.
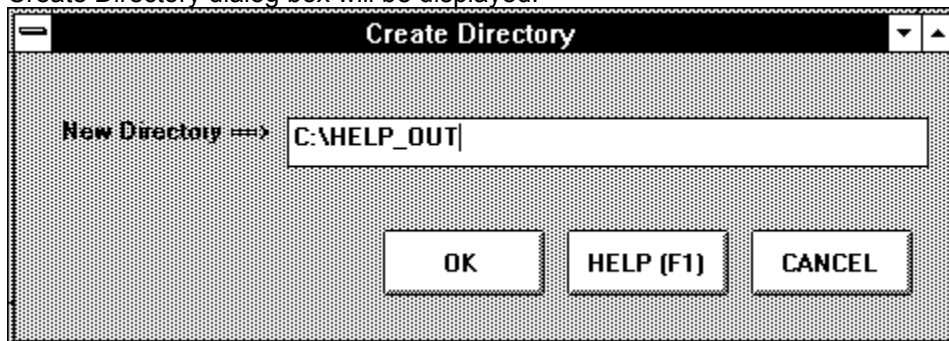
99% of the time, normal VB projects are clean after one or two iterations.

Selecting the Auto-Rewrite Path

Click the SELECT PATH button on the main SCOPER form after the analysis phase completes. The Select Path dialog box will be displayed as shown below:
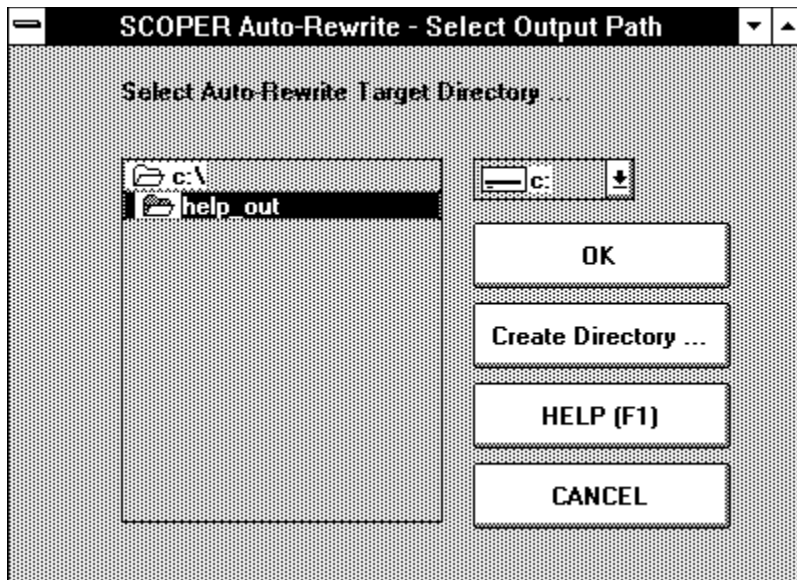

If you want to create a new directory for the auto-rewrite output, click the Create Directory button. The Create Directory dialog box will be displayed:
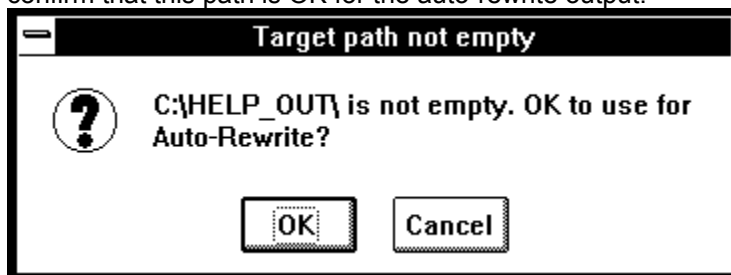

In the above example we will create a directory called C:\HELP_OUT. Click OK to create the new directory.

Use standard Windows techniques to select the drive and path for the rewrite output. In the example below, we have selected C:\HELP_OUT as the rewrite path:

SCOPER will not allow you to select the path of the current project as the rewrite path. (Try it!). Also, if the path you select is not empty (i.e., if there are already some files in the directory) SCOPER will ask you to confirm that this path is OK for the auto-rewrite output.

Note that if you say OK in this situation, SCOPER will not delete any files currently in the directory. Thus, your rewritten source will reside in a directory that contains other files that may not be involved in your VB project.

Click OK once you have selected the auto-rewrite path. The SCOPER main form will now appear as follows:

C:\AA_TEST\PROJECT1.MAK
Unused Elements ...

	Element Name	Type	Scope	Routine	
1	GetFreeSystemReso	Declare	Module	Declarations	C:\AA_TEST\FORM1.FRM
2	Mod_const	Const	Module	Declarations	C:\AA_TEST\FORM1.FRM
3	pnLocalInt	Integer	Local	Form_Load	C:\AA_TEST\FORM1.FRM
4	gcTest	Const	Global	Declarations	C:\AA_TEST\MODULE1.E
5	gcTest2	Const	Global	Declarations	C:\AA_TEST\MODULE1.E
6	gnTest	Integer	Global	Declarations	C:\AA_TEST\MODULE1.E
7	gnTest2	Integer	Global	Declarations	C:\AA_TEST\MODULE1.E
8	int_a	Integer	Global	Declarations	C:\AA_TEST\MODULE1.E
9	ModConst	Const	Module	Declarations	C:\AA_TEST\MODULE1.E
10	second_sub	Sub	Global	second_sub	C:\AA_TEST\MODULE1.E

SELECT Project ...   FILTERS ...   PROTECT   UNPROTECT

OPTIONS ...   SELECT PATH ...   PRINT   STATS   HELP (F1)

SCOPER   AUTO-REWRITE   SAVE TO FILE ...   CROSS-REF.   EXIT

Rewrite Path ==> C:\HELP_OUT\
Module Progress: ▓▓▓▓▓▓▓ (0)% ▓▓▓▓▓▓▓   Project Progress: ▓▓▓▓▓▓▓ (0)% ▓▓▓▓▓▓▓

Click AUTO-REWRITE to begin

Notice that the AUTO-REWRITE button is now enabled. Directly below the AUTO-REWRITE button the caption 'Rewrite Path ==>' is displayed followed by the rewrite path you have selected. In the above example we see that our rewrite path will be C:\HELP_OUT\. Make sure the rewrite path is correct before proceeding.

Starting Auto-Rewrite

Once you are sure that the rewrite path is correct, click the AUTO-REWRITE button.

SCOPER begins by sorting the spreadsheet of unused elements for the rewrite process. The message bar will display 'Sorting for rewrite.' If your project has a high volume of unused elements, this sort may take a few moments.

Note that SCOPER will *only remove unused elements that appear in the spreadsheet* during the auto-rewrite process. If you have used the FILTERS ... form to filter some type/scope combinations or have protected individual elements using the PROTECT button, SCOPER will not remove those unused elements from the project during the auto-rewrite process. Simply put, if an element is not in the unused element spreadsheet it will be left alone during auto-rewrite.

After the spreadsheet of unused elements has been sorted, SCOPER copies the entire contents of the current project directory into the rewrite target path. The message bar will display 'Loading {rewrite path}' during this process. In the above example the message would be 'Loading C:\HELP_OUT\'. This proces should complete in a few seconds.

Once the rewrite target path is loaded, SCOPER will begin rewriting the code in the target path directory. The message bar will display '{file name} - Rewriting' for each source code file that SCOPER needs to work on. For an average size VB project, the rewrite process takes about 5 seconds.

When the rewrite process is complete, the message bar will display 'Rewrite complete'

Related Topics:

The New Project .MAK File

The New Project .MAK File

SCOPER creates the new project .MAK file in the auto-rewrite target directory as follows:

If a source file outside the original project directory had to be rewritten to eliminate dead code (e.g., C:\VB\CONSTANT.TXT) SCOPER will modify the new .MAK file to point to the rewritten CONSTANT.TXT file in the auto-rewrite target directory.

If the original project .MAK file contained any hard-coded references to the original project directory, SCOPER will modify these references in the new project .MAK file to point to the auto-rewrite target directory.

Printing

After <u>SCOPER</u> has finished analyzing your <u>project</u>, the PRINT button on the main SCOPER form will be enabled if any <u>unused elements</u> were found. <u>Click</u> PRINT to obtain a hardcopy image of the unused element spreadsheet.

Note that the hardcopy unused element report will be sequenced the same way as the spreadsheet (filename, routine, and element name).
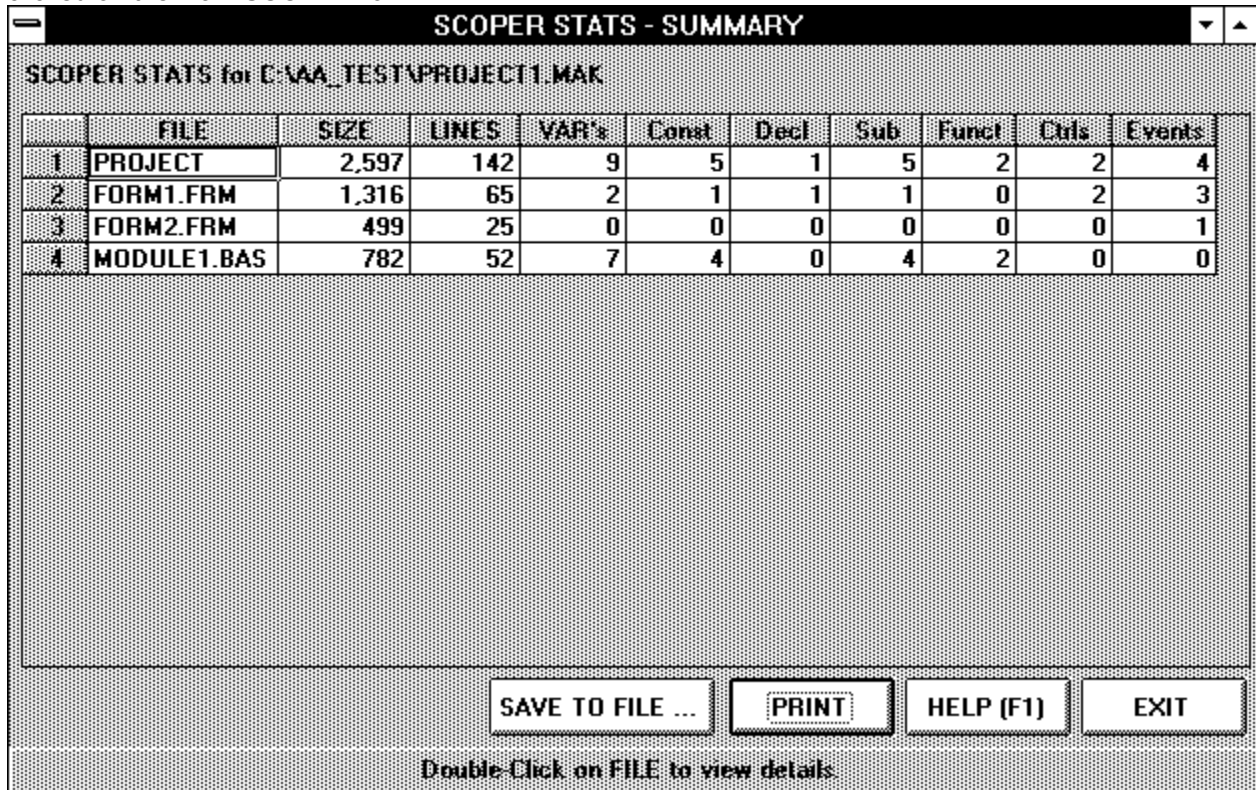
Save to File

After <u>SCOPER</u> has finished analyzing your <u>project</u>, the SAVE TO FILE button on the main SCOPER form will be enabled if any <u>unused elements</u> were found. <u>Click</u> SAVE TO FILE to create a <u>tab-delimited</u> file of the unused elements in your project.

SCOPER will prompt you for a target filename. Once created, the file is suitable for importing into Excel, etc. where you can re-sort the data if you wish and run any other types of analysis you desire.

Statistics Summary Form

The SCOPER Statistics Summary Form (see example below) is displayed when the STATS button is clicked on the main SCOPER form.

	FILE	SIZE	LINES	VAR's	Const	Decl	Sub	Funct	Ctrls	Events
1	PROJECT	2,597	142	9	5	1	5	2	2	4
2	FORM1.FRM	1,316	65	2	1	1	1	0	2	3
3	FORM2.FRM	499	25	0	0	0	0	0	0	1
4	MODULE1.BAS	782	52	7	4	0	4	2	0	0

SCOPER STATS - SUMMARY

SCOPER STATS for C:\AA_TEST\PROJECT1.MAK

SAVE TO FILE ...   PRINT   HELP (F1)   EXIT

Double-Click on FILE to view details

On the first row, statistics are displayed for the entire project. Starting with the second row, statistics are displayed for each individual file in the project.

The SAVE TO FILE and PRINT buttons work the same way as the corresponding buttons on the main form.

Related Topics:

Drilling Down

Drilling Down

More detailed statistics can be displayed as follows. Position the mouse pointer anywhere on the row of statistics you wish to examine further and double-<u>click</u>. In this example, double-clicking on the <u>project</u> row will display the following drill-down information:

	LOCAL		MODULE		GLOBAL		TOTAL	
	COUNT	UNUSED	COUNT	UNUSED	COUNT	UNUSED	COUNT	UNUSED
Integer	4	2	1	0	3	3	8	5
Long	0	0	0	0	0	0	0	0
Float	0	0	0	0	0	0	0	0
Double	0	0	0	0	0	0	0	0
Currency	0	0	0	0	0	0	0	0
String	1	0	0	0	0	0	1	0
Variant	0	0	0	0	0	0	0	0
User-Type	0	0	0	0	0	0	0	0
Const	1	1	2	2	2	2	5	5
Declare's	0	0	1	1	0	0	1	1
Function (C)	0	0	0	0	2	1	2	1
Sub (C)	0	0	1	0	4	2	5	2
Totals	6	3	5	3	11	8	22	14
Source Lines:	142	Bytes:	2,597					

SCOPER STATS - DETAILS

DETAILED STATS for C:\AA_TEST\PROJECT1.MAK   (PROJECT)

SAVE TO FILE ... | PRINT | HELP (F1) | EXIT

As this example shows, <u>SCOPER</u> can drill-down to the detail level for the entire project or for any source module within the project.

The SAVE TO FILE and PRINT buttons work the same way as the corresponding buttons on the main form.

Enabling the Cross-reference

THIS FEATURE IS AVAILABLE TO REGISTERED USERS ONLY.

The SCOPER cross reference is enabled by checking the 'SAVE DATA FOR CROSS-REF. option before starting the analysis phase. Keep in ming that instructing SCOPER to build cross-reference data while it is examining your project source code will increase the run time of the analysis phase by approximately 40%

Displaying Cross-reference Data

If you checked the 'SAVE DATA FOR XREF' option, the 'CROSS-REF' button on the main SCOPER form will be enabled when the analysis phase completes. Click 'CROSS-REF' to enter he SCOPER cross reference sub-system.

The message 'Sorting for Cross-Ref.' while SCOPER sorts the cross data saved during the analysis phase. This may take a minute or two for large projects.

When you initially enter the Cross-Reference screen, SCOPER will ask you to check the display filters and then click LOAD.

Related Topics:

Display Filters

Display Filters

The SCOPER Cross-Reference display filters allow you to customize the cross-reference report to include only the elements you want. If you leave all of the boxes checked, you will see a complete cross reference of every single element in your project.

If you want to 'filter' your cross reference report, remove some of the checks. In the example below, we have used the filter to limit the cross reference report to variables and constants:

In the above example, row #5 tells us that there is a module scope integer named int_a defined on line 34 in the Declarations section of FORM1.FRM.

Row #6 shows that int_a is referenced on line 6 of the Form_Load event in FORM1.FRM. This reference is at absolute line# 49 of FORM1.FRM.

Row #7 tells us that int_a is also referenced on Line 3 of the Text1_Change event in FORM1.FRM. This reference is at absolute line #62 of FORM1.FRM.

Row 8 shows another integer named int_a defined in the Declarations section of MODULE1.BAS on Line #3. This int_a has global scope and is unreferenced throughout the project.

The filter options can be changed at any time to any desired combination of scope and type. If you change the filter options, remember to click LOAD to reload the spreadsheet.

The SAVE TO FILE and PRINT buttons work the same way as the corresponding buttons on the main form.

Phone

To order by phone, have your credit card ready and call:

1-800-447-ISES

The ISES office is open from 10 AM to 4 PM (Eastern Time) Monday thru Friday.

Credit Card

To order by credit card, print or photocopy this form, fill it out, and fax the completed form to 908-580-1008 or mail it to:

ISES Inc., 102 Sunrise Drive, Gillette, NJ 07933

Your Name: _____

Address: _____

                          _____

City: _____

State: _____

Zip: _____

Country: _____

SCOPER B.10 Single User: ____ copies @ $19 =   _____

SCOPER site license:

      2 to 9 computers: ____ computers @ $16 =   _____

      10 to 24 computers: ____ computers @ $14 =   _____

      25 to 49 computers: ____ computers @ $12 =   _____

      50 to 99 computers: ____ computers @ $10 =   _____

               Unlimited site license @ $1,000 =   _____

               NJ residents add 6% sales tax +_____

Outside   US, add shipping & handling (see chart):   _____

                          Total = _____

Master Card ___   Visa ___   American Express ___

Card# _____ Exp Dt _____ Sig: _____

SWREG

A single user registered copy of <u>SCOPER</u> can be ordered via SWREG on Compuserve.

The SCOPER SWREG ID is 5904.

Simply type GO SWREG at the   Compuserve ! prompt and follow the menus.

CHECK

To order by check, print or photocopy this form, fill it out, and mail the completed form with a check payable to ISES, Inc. to:

ISES Inc., 102 Sunrise Drive, Gillette, NJ 07933

Name: _____

Address: _____

               _____

City: _____

State: _____

Zip: _____

Country: _____

SCOPER B.10 Single User: ____ copies @ $19 =   _____

SCOPER site license:

       2 to 9 computers: ____ computers @ $16 =   _____

       10 to 24 computers: ____ computers @ $14 =   _____

       25 to 49 computers: ____ computers @ $12 =   _____

       50 to 99 computers: ____ computers @ $10 =   _____

                Unlimited site license @ $1,000 =   _____

                NJ residents add 6% sales tax +_____

Outside   US, add shipping & handling (see chart):   _____

                              Total =   _____

Shipping and Handling

Shipping and handling charges for orders outside the US are:

Canada and Mexico	$	3.00
Europe		5.00
South America		6.00
Africa		8.00
Asia		8.00
Middle East		9.00
Pacific Rim		9.00
Australia / New Zealand		9.00

CompuServe

If you require technical assistance or wish to request enhancements or improvements to future versions of SCOPER, the fastest way to contact us is via compuserve e:mail (GO MAIL). Send your query with source code examples (if possible) to 72417,627.

America On-Line

Any technical issues can be e:mailed to us on AOL. Send to ISES INC

Fax

You can reach us by fax at 908-766-1109 or 908-580-1008. Please write SCOPER on your fax header sheet. Please try to include as much information as possible in your query, including source code if possible.

Mail

You can reach us by mail at :

ISES, Inc.   102 Sunrise Drive,   Gillette, NJ   07933

Please try to include as much information as possible in your query, including source code if possible.

Voice

Our technical support voice line is 908-766-1894. We recommend using Compuserve or Fax services, however, as it will save you money and give us a chance to study your query more thoroughly.

Glossary of Terms

Analysis Phase

Auto-Rewrite

click

Cross-Reference

Dead Code

Declarations

development cycle

dialog box

Drill Down

event

Filter

Iteration

MAK

Message Bar

Parsing

pass

Path

Progress bar

project

project directory

Recursion

scope

SCOPER

shareware

tab-delimited

unused elements

Unused Elements Spreadsheet

VB

Analysis Phase

Part of processing when <u>SCOPER</u> is identifying <u>dead code</u> in a <u>VB</u> <u>project</u>.

Auto-Rewrite

SCOPER process of rewriting VB source automatically eliminating dead code.

click

Windows term describing the process of pressing the left mouse button down and then letting it up.

Cross-Reference

Source code report showing <u>project</u> elements, where they are defined, and where they are used.

Dead Code

Variables, constants or routine <u>declarations</u> that are never referenced. Functions and Subroutines that are never called.

Declarations

In <u>VB</u> terminology refers to the section of a source module before the first Function, Subroutine, or <u>Event</u>.

development cycle

The life-cycle of the software creation process.

dialog box

Windows term describing a small pop-up window designed to ask a specific question of the user and return the answer to the main program.

Drill Down

Finer level of detail about a particular statistic.

event

An action recognized by an object, such as clicking the mouse or pressing a key, and for which you can write code to respond. Events can occur as a result of user action or program code, or they can be triggered by the system.

Filter

Enables selection of a subset of objects from the overall population based upon some common object attribute.

Iteration

Process of repeating a step or series of steps. Also known as a 'loop' in programming circles.

MAK

Refers to a <u>VB</u> <u>project</u> file where all project components are identified.

Message Bar

Panel on the bottom of SCOPER forms displaying important processing information to the SCOPER user.

Parsing

The process of seperating a computer instruction into its smallest components or tokens. Also referred to as tokenizing.

pass

Term used in compiler technology describing the process of reading and <u>parsing</u> programming source code.

Path

Directory structure definition leading to a target for data files.

Progress bar

Popular method of graphically displaying program execution status to the user. Usually fills from left to right with a percentage complete display in the middle.

project

In <u>VB</u> terminology, refers to the set of source modules used to create a piece of software.

project directory

The directory where the project .MAK file exists.

Recursion

Programming technique where a function calls itself. Also applies to two different functions that call each other.

scope

Term used to describe the realm of a variable, constant, or routine. Possible values are 'global', 'module', and 'local'.

SCOPER

Essential add-on tool for serious <u>VB</u> programmers. Primary function is to eliminate <u>dead code</u> from VB projects.

shareware

Popular software distribution technique where users get to try a software product before purchasing a registered copy.

tab-delimited

A file where the fields are seperated by tab characters (Chr$(9)). Suitable for quickly importing into Excel, etc.

unused elements

Variables, constants or routine <u>declarations</u> that are never referenced. Functions and Subroutines that are never called.

Unused Elements Spreadsheet

Spreadsheet on the main <u>SCOPER</u> form containing <u>dead code</u> found during the <u>analysis phase</u>.

VB

Visual Basic 3.0 Professional Edition for Windows.